2013-06

Electrical ship demand modeling for future generation warships

Sievenpiper, Bartholomew J.

Monterey California. Naval Postgraduate School

http://hdl.handle.net/10945/40262

Electrical Ship Demand Modeling for Future Generation Warships

by

Bartholomew J. Sievenpiper
B.S. Chemical Engineering,
University of Pittsburgh 2002

Submitted to the Department of Mechanical Engineering in Partial Fulfillment of the Requirements for the Degrees of
Naval Engineer
and
Master of Science in Mechanical Engineering
at the
Massachusetts Institute of Technology
June 2013

Signature of Author...[redacted]
Department of Mechanical Engineering
May 10, 2013

Certified by...[redacted].................
Steven B. Leeb
Professor of Electrical Engineering & Mechanical Engineering
Departments of Electrical Engineering & Mechanical Engineering
Thesis Supervisor

Certified by......[redacted].........................
Norbert H. Doerry
Technical Director
Technology Office, Naval Sea Systems Command
Thesis Supervisor

Accepted by..[redacted]
David Hardt
Chairman, Department Committee on Graduate Students
Department of Mechanical Engineering

Electrical Ship Demand Modeling for Future Generation Warships
by
Bartholomew J. Sievenpiper

Submitted to the Department of Mechanical Engineering in Partial Fulfillment of the Requirements for the Degrees of

Naval Engineer

and

Master of Science in Mechanical Engineering

Abstract

The design of future warships will require increased reliance on accurate prediction of electrical demand as the shipboard consumption continues to rise. Current US Navy policy, codified in design standards, dictates methods of calculating the average demand power. Using several modern sources of information for the DDG-51 class ship, this thesis investigates the utility of current analysis techniques and examines possible improvements. This thesis expands upon a basic method of modeling and simulation to develop a design tool that would provide an improved method of predicting ship electrical loads with increased fidelity of the ship's electrical demand. These efforts ultimately allow a better understanding of ship behavior to enable decision making in all stages of Navy ship design.

Thesis Supervisor: Steven B. Leeb
Title: Professor of Electrical Engineering and Mechanical Engineering

Thesis Supervisor: Norbert H. Doerry
Title: Technical Director, NAVSEA SEA 05 Technology Office

Acknowledgements

Special thanks are warranted for the many people who helped make completion of this possible. The support of Steve Markle at PMS-320 was critical in getting this project off the ground, and to Steve Leeb for exploring the many ways the DDG-51 data set could be utilized. The support of the crews of the ships and Navy facilities we visited was beyond what we could have hoped, and ensured the project was successful.

Additionally, special thanks to the research partners/ travel companions for these efforts: Travis Anderson, Katie Gerhard, and John Donnal. Thanks Uzoma Orji for assistance in building the models that form a framework of this thesis.

Finally, my greatest appreciation is to my family for supporting me in all these endeavors. Traci, Max and Josef, your support made it possible to get this work done.

Table of Contents

Table of Figures

Table of Tables

1 Motivation and Background

Electrical power generation systems for naval vessels have become increasingly complex in recent years. Interest in the application of tools for the design and control of electrical micro-grids has increased in recent years as means of providing improved efficiency and reliability of power delivery. For the US Navy the DDG-51 is the largest and most recognizable combatant ship class at sea today, and its power system represents a canonical example of a shipboard micro-grid. Using the DDG-51 power distribution system as an exemplar for the challenges and opportunities faced in the coming decades, this thesis examines current power system design methods and provides a framework for improved future design tools.

1.1 Motivation for Research

The design process for Navy ships is complex, requiring decisions in the present for ships that will be operating several decades into the future. Onboard electrical generation plants have seen tremendous growth over the course of the 20th and 21st centuries, a trend that shows no signs of abating. In a recent study performed by the US Navy on alternative propulsion methods by Webster (et al), the authors examined the maximum margined electrical load for ships. The resulting historic and projected electrical load growth, driven primarily by combat systems load growth, is shown in Figure 1 [1].

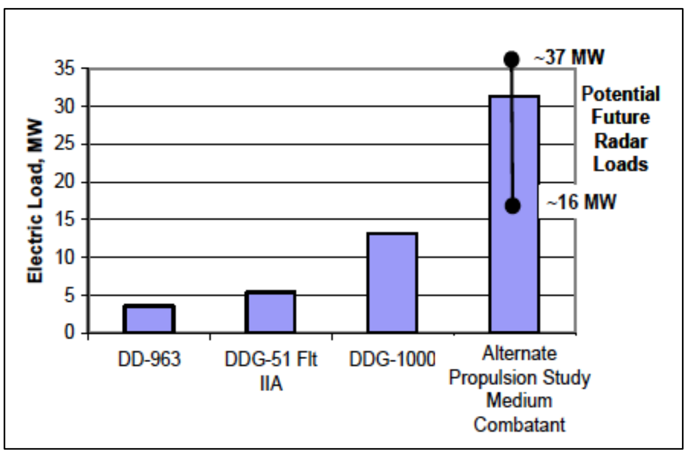

Figure 1: Electrical Load Growth on Surface Combatants [1]

At the same time the study on alternative propulsion methods was being conducted the Navy, through Naval Sea Systems Command (NAVSEA), was producing a technology roadmap for the next generation integrated power system (NGIPS) [2]. An integrated power system (IPS) in the Navy refers to a system in which the ship's service electrical distribution and ship's propulsion power are provided by a single distribution system. The purpose of the NGIPS development is to understand technical challenges and the enabling technologies required to move to an all-electric warship. Demonstrated graphically, the roadmap is shown in Figure 2.

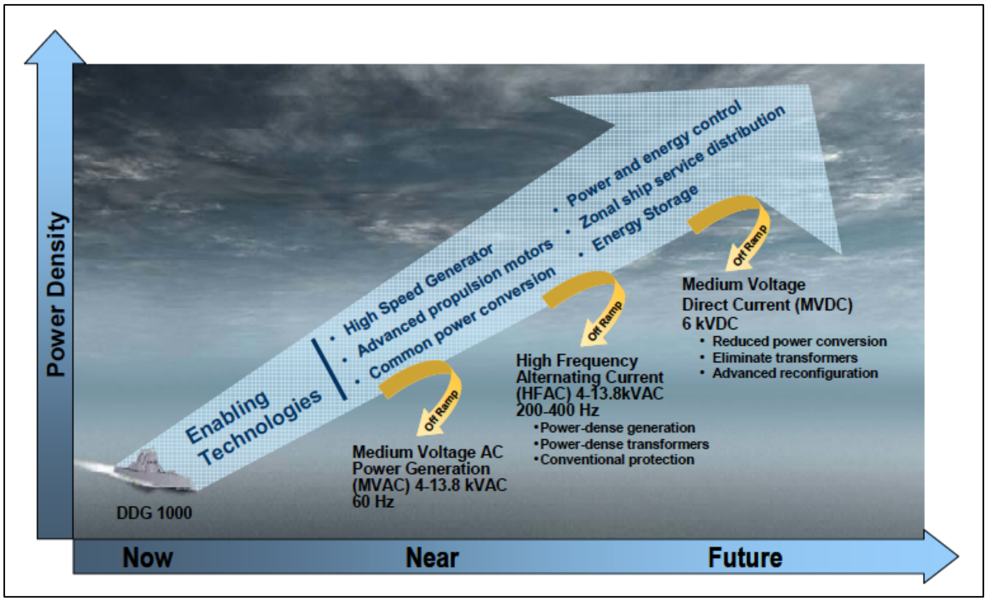

Figure 2: NGIPS Technology Development Roadmap[2]

A limitation called out in the NGIPS process, however, is that the current methods used to size and characterize shipboard loads may be insufficient for these future electrical distribution methods [2]. Defining the loads is critical in designing and sizing major electrical generation components.

The design guidance for determining electrical plant sizing calls for first developing an electric power load analysis (EPLA). This guidance is codified in the data design sheet (DDS) 310-1 [3], published by NAVSEA. The EPLA is a crucial portion of the design of a

ship. It is utilized to determine component sizing in the electrical distribution system for everything from generating capacity to breaker and cable sizing. One limitation presented by this method exists, however, in that these have been historically performed as without a standardized process [4]. Additionally, the EPLA is not presented in a manner that supports answering dynamic questions, such as quality of service [5].

Developing an ship electrical model that is flexible enough relevant to be throughout different stages of the ship's life cycle would address both current and future needs. Such a model would be capable of not only producing the a single value for the expected shipboard load but could be queried to show changes output based on varying input parameters. Instead of functioning merely as a tool to size electrical components, it could be used to develop dynamic fuel consumption estimates or perform sensitivity analyses based on changing fleet operational needs. A program developed that is extensible on both the input and output side would provide the entire design community a tool useful for answering problems not yet envisioned. This thesis will develop the framework for such a model by using the DDG-51 class to examine current methods and define one possible process to build a next-generation model.

1.2 Why Focus on the DDG-51 Class?

The DDG-51 class (or ARLEIGH BURKE class) is simultaneously the largest current class of ships in the Navy's inventory and a large portion of planned acquisition in the coming years. To date, there have been 62 DDG-51 class ships commissioned, with three major class variants. These variants are known as the Flight I, II, and IIA ships. The Navy's 30-year shipbuilding plan expects a fourth variant, the Flight III, to enter service in the early 2020's [6]. These naval assets will therefore be constructed and utilized well into the future. The expected inventories of surface combatants over coming decades will be dominated by the DDG-51 class, as can be seen in Figure 3.

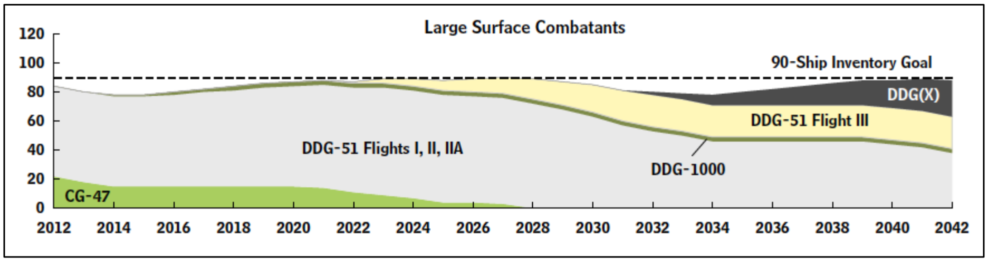

Figure 3: Expected Inventory of Large Surface Combatants for Next 30 Years [6]

The quantity and length of production for these ships make it a unique platform to study for several reasons. The construction of the Flight III version will require design work in the coming years for which a large body of data already exists. The ability to fully understand the manner in which the current DDG-51s are operated gives a potential advantage to the ship design community, as fewer assumptions are required in the design.

Another potential benefit of studying this class of ships is that the large number in operation makes them a fertile research area for technology changes or enhancements. Systems such as hybrid electric drive (HED) are currently being developed and tested for potential inclusion into the propulsion train of future or existing destroyers[7]. With the HED system the ship would utilize a motor, powered by the ship's electrical distribution system, at slow speeds as a means of saving fuel [7]. Other recent studies have examined hydrodynamic changes such as larger or contra-rotating propellers, a bow bulb, a stern bulb, or an updated stern flap [8]. Each of these hydrodynamic changes would be performed as a means of lowering fuel consumption over the operational lifetime of the ship. These studies also demonstrate that the large size of the ship class justifies the investment in design changes. Small savings applied over many ships can result in large cost avoidance for the Navy.

One additional reason to consider the DDG-51 as an excellent target for study is that it has a large number of onboard sensors monitoring equipment. The growth of remote sensing and control onboard ships has increased in recent designs, as sensors have become more

prevalent. The ability to collect data for future analysis using onboard functionality is a capability that was exploited to gather data in this thesis.

1.3 DDG-51 Class Ship Introduction

The ARLEIGH BURKE class destroyers referred to in this paper are modern warships capable of fulfilling missions ranging from anti-air warfare (AAW) to anti-submarine warfare (ASW) to anti-surface warfare (ASUW). To accomplish this variety of missions, the ships are constructed with a wide variety of weapons and detection systems. Weapons systems include a 5-inch gun, a variety of air, land, and anti-missile missiles, torpedoes, and small arms. Detection systems range from SPY-1D 3-D phased array radar to surface search radar to the SQQ-89 sonar [9]. This wide variety of equipment onboard ensures that the DDG-51 possesses tremendous operational flexibility. A summary of the basic characteristics of the DDG-51 class is shown in Table 1.

Table 1: DDG-51 Class Basic Characteristics

DDG Characteristics			
	Flight I	Flight II	Flight IIA
Length (ft)	505	505	509
Beam (ft)	59	59	59
Draft (ft)	30.5	30.5	30.5
Displacement (LT)	8320	8673	9496
Manning	276	276	276
Speed (kts)	30+	30+	30+
Shafts	2	2	2
Gas Turbines (Per Shaft)	2	2	2

To fully understand the impact of the information presented later in this thesis, it is important to give a basic functional description of the engineering plant operating modes for the DDG-51 class ship. The DDG-51 has two main machinery rooms, each of which independently possesses the necessary equipment to power one shaft. Each shaft has by two gas turbine modules (GTM), each of which contains a GE LM 2500-30 gas turbine engine capable of generating approximately 27000 shaft horsepower (SHP). Each pairs of GTMs are mated to their associated shaft through a main reduction gear, taking the high speed turbine output and producing a lower speed high torque shaft rotation (1A and 1B

GTM power the starboard shaft, 2A and 2B GTM power the port). Each shaft is equipped with a controllable pitch propeller (CPP), and is operated with programmed logic to select an optimum pitch and shaft rotational speed for a given ordered speed through the water.

When this paper discusses the engineering modes of the ship, it is referring to the selection of the propulsion train configuration for the vessel. The **trail shaft** engineering mode is a configuration where one shaft is being powered by a single GTM while the remaining shaft is allowed to spin free. The trail shaft configuration is the most efficient engineering mode in terms of fuel consumption, at the cost of a restricted top speed. Additionally, the trail shaft mode uses only a single GTM, which presents a potential operational risk in the event of an engine casualty. The reliability concern and speed restriction can be partially addressed using the **split plant** engineering mode. The split plant mode consists of two operating GTMs, with one powering each shaft. Split plant allows greater top speed (though still limited), and increased propulsion reliability when desired by ship operators. The **full power** operating mode for the engineering plant is the condition where all 4 GTMs are operated, with 2 GTMs powering each shaft. While the full power mode allows the ship to travel at its maximum speed of 30+ knots, it is the least fuel efficient method of operating the ship. The full power mode also represents the maximum amount of propulsion redundancy for the ship, and is typically utilized when the potential impact of a propulsion casualty could lead to catastrophic effects for the ship.

The electric plant for the DDG-51 class has some variation depending on the flight of the vessel in question. All DDG-51 class ships generate electrical power using 3 Allison 9140 gas turbine generators (GTG). The earlier GTG sets were rated to 2500 kW, with later ships being built with 3000 kW generator sets. The typical operating configuration for the DDG-51 is to have 2 of 3 GTG sets online. The original Flight I design of the DDG-51 electrical plant utilized a radial distribution system, which was fully replaced by a zonal electrical distribution system (ZEDS) for ship hull numbers 78 and higher. The ZEDS architecture is designed to distribute electrical power within specific zones to ensure the ship is better prepared to retain fighting capability following potential damage. The ZEDS breaks the ship down into 6 primary zones, and with buses serving the upper and lower portions of

the ship [10]. The electrical generation system also utilizes multi-function monitors (MFM) to provide fault isolation for the distribution system. A pictorial representation of all major switchboards, and the MFM monitoring them, is shown in Figure 4.

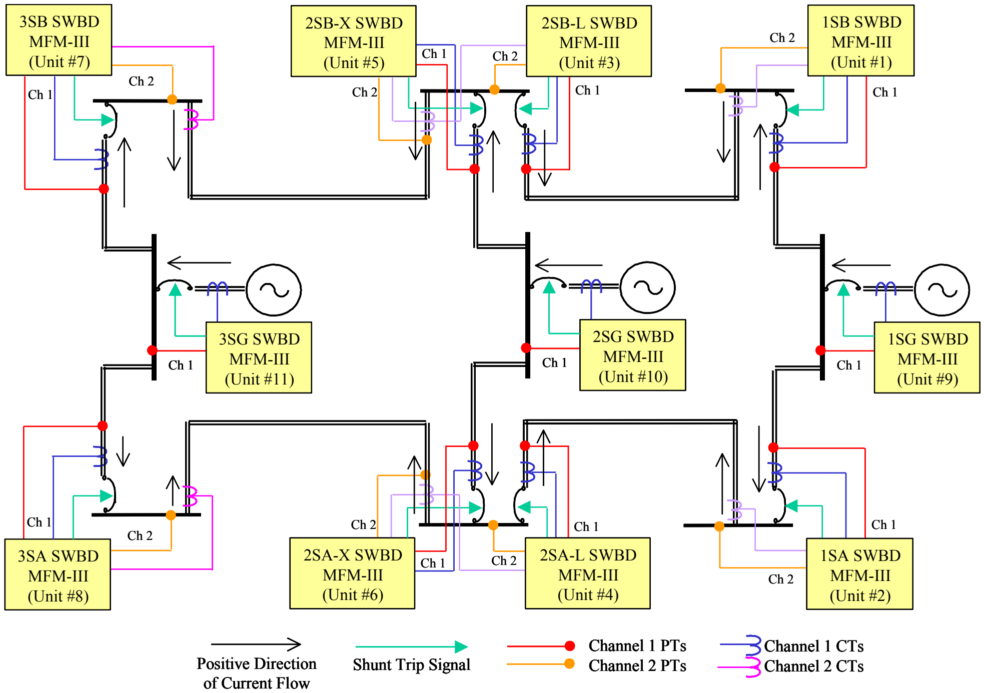

Figure 4: ZEDS Electrical Distribution (Switchboards and MFMs Shown) [11]

1.4 Electric Plant Load Analysis (EPLA)

At the heart of the design process for the electrical distribution system is the EPLA, which itself is simply a product of estimated electrical load on the ship. This analysis of demand load is then used to perform a variety of other tasks in the ship design process, ranging from estimating fuel consumption to sizing electrical components. An example of these different tasks, and the sections of DDS 310-1 they are defined in, is presented in Figure 5.

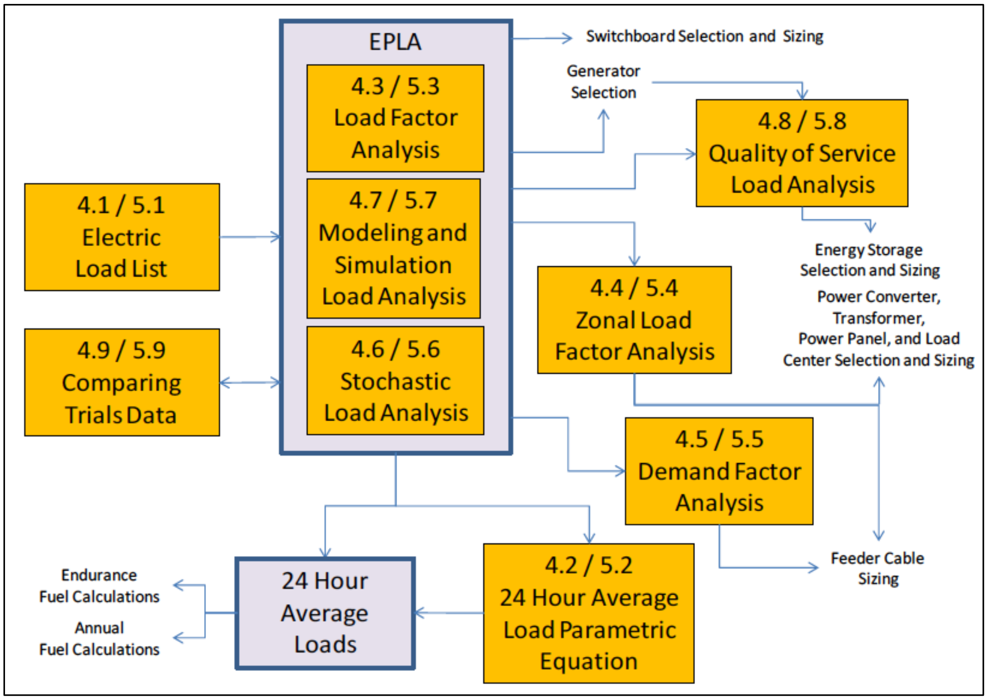

Figure 5: Inter-relationship of DDS 310-1 Tasks [3]

Fundamentally, the EPLA uses the list of all loads installed onboard the ship and utilizes one of three methods for calculating demand power. Each of these methods has benefits and drawbacks, and will be further described in the following sections.

1.4.1 Load Factor Analysis

Load factor analysis is the means that has historically been utilized for the performance of a ship's EPLA. This method assumes that each load onboard a ship is small relative to both the prime mover's rating and the operating load of the ship. In this method of performing an EPLA, each onboard load is assigned a load factor for a given set of conditions. The load factor represents the long-term average operating power level as a fraction of the component's rated load. As a result, to calculate a load factor one must estimate the fraction of the time a system will be in operation and the average power the component will draw when in operation. The load factor will be the product of these values, as shown in Equation 1.

$$LF = (\boldsymbol{Fraction\ of\ Operation}) \cdot \frac{\boldsymbol{Average\ Power\ Consumed}}{\boldsymbol{Rated\ Power}}$$

Equation 1: Load Factor Calculation

When the load factor for a given load is being calculated it is important to note that most components will not operate at rated load. Since most pumps and motors are purchased in standard sizes and not custom designed for a given purpose these items are often oversized for a specific use. As a result, components will typically not operate at full power even when a system is operating at maximum design loading conditions. Determining the expected operational fraction or average power consumption might be performed using historical data from similar types of ships or systems or by using manufacturer design information such as pump curves. When information is not available, standard estimated values for load factors are provided in DDS 310-1.

The above description for a load factor must be applied to a variety of operating states for the ship. The four conditions that the ship is analyzed in when performing an EPLA are shore, anchor, cruising, and functional. The shore condition is the period of time when the ship is in port and shut down and electrical power is supplied by a tender or shore power. Anchor refers to the condition where the ship is moored or anchored, but supplying its own electrical power. Cruising is the condition in which the ship is sailing with defense capability, but is not engaged in its functional mission. The functional condition refers to the period of time when the ship is performing its primary mission. For surface combatants (including the DDG-51) this would be a period of battle, though for other ship classes it varies based on the functional mission of the ship. The additional condition of emergency may be included for a ship possessing separate emergency and ship's service electrical generation (i.e. a steam-powered ship), and would describe the state where the ship is underway operating only on the emergency generator. This does not apply to the DDG-51 class, and will not be discussed further in this thesis.

For each of these operating conditions, the ambient external conditions must also be evaluated. This is performed by assigning a load factor for summer and winter under each

of the four conditions discussed above. For some systems, there may be little variation based on the environmental state. System components involved heating and cooling are most impacted by these external effects.

For each component on the electric load list the load factors are tabulated and then used to compute a calculated load for each condition. The calculated load is the product of the load factor and the rated load, as shown in Equation 2.

$$\boldsymbol{Calculated\ Load = LF \cdot P_{Rated}}$$

Equation 2: Calculated Load

An example of the tabulated results for two nominal components is given in Table 1. In this example, a pump is envisioned that operates infrequently while moored, but at increasing rates when in the cruising and functional condition. A ventilation heater also is represented, but is assumed dependent only on the ambient season, not on the operating condition.

Table 2: Load Factor Analysis Calculation

Component	Rating		Shore				Anchor			
			Summer		Winter		Summer		Winter	
	kW	Hp	LF	Load	LF	Load	LF	Load	LF	Load
Pump	65	50	0.1	6.5	0.1	6.5	0.1	6.5	0.1	6.5
Ventilation Heater	50	--	0	0	0.5	25	0	0	0.5	25

Component	Rating		Cruising				Functional			
			Summer		Winter		Summer		Winter	
	kW	Hp	LF	Load	LF	Load	LF	Load	LF	Load
Pump	65	50	0.6	65.6	0.6	65.6	0.9	58.5	0.9	58.5
Ventilation Heater	50	--	0	0	0.5	25	0	0	0.5	25

To complete the load factor analysis method of the EPLA, all components onboard the ship would be included in tables such as this. The columns are then summed to generate the total expected load onboard the ship in each operating condition and ambient state for the ship, resulting in eight different EPLA load estimates for the ship. Tabulated estimates for load in specific load centers or switchboards could be performed in a similar manner to aid in sizing these portions of the electrical distribution system.

While there are several benefits of using load factor analysis, there are also limits to its utility. It is a straightforward way to estimate the electrical demand on the ship, and by using historically defined load factors this estimate can be quickly achieved. For many applications, such as defining the 24-hour average electrical demand to determine annual fuel consumption, this method may provide a reasonable estimate. The load factor analysis does not give indications of maximum expected peak power demand, however, as it creates long-term averages instead of loading at specific points in time. To compensate for these effects, the use of electrical margins must be incorporated into a design to ensure the generation capacity is not undersized.

1.4.2 Stochastic Load Analysis

The stochastic load analysis is an alternative method provided in DDS 310-1 for estimating the demand power onboard a ship in the design process. This method assumes that for each component a probability distribution function (PDF) and associated cumulative distribution (CDF) for electrical loading can be determined or estimated. Examples of the three most typical distributions expected for shipboard modeling are the uniform, triangular, and discrete distributions, shown in Figure 6, Figure 7, and Figure 8 respectively. Care must be chosen when implementing a distribution to ensure it reflects the actual loading conditions and real-world possibilities. Using a distribution with infinite tail sections, such as a normal distribution, could create negative loading conditions, and must be judiciously applied.

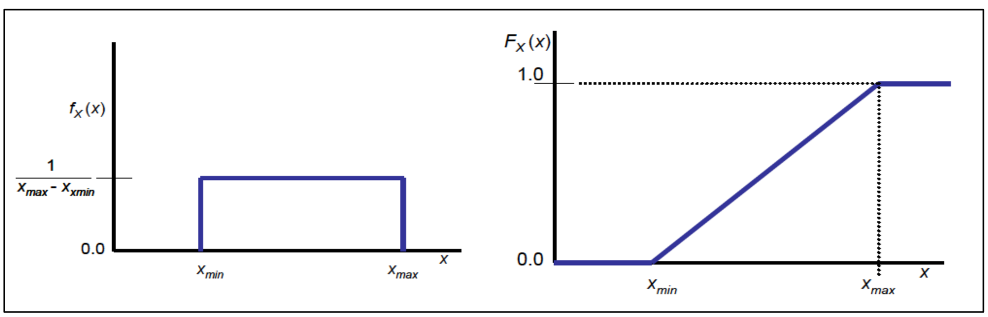

Figure 6: Uniform Distribution PDF and CDF [3]

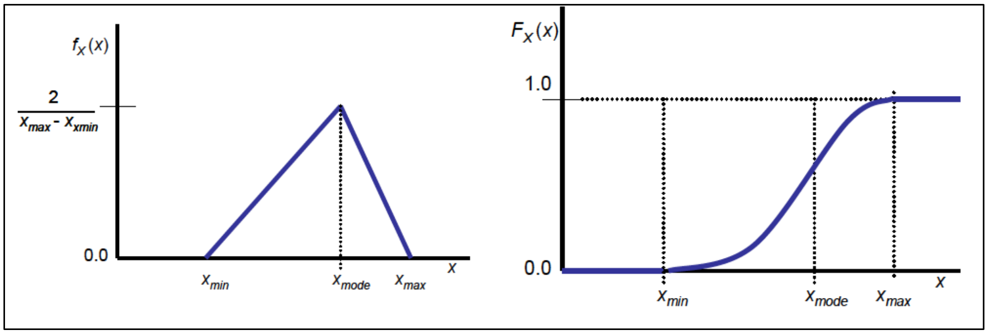

Figure 7: Triangular Distribution PDF and CDF [3]

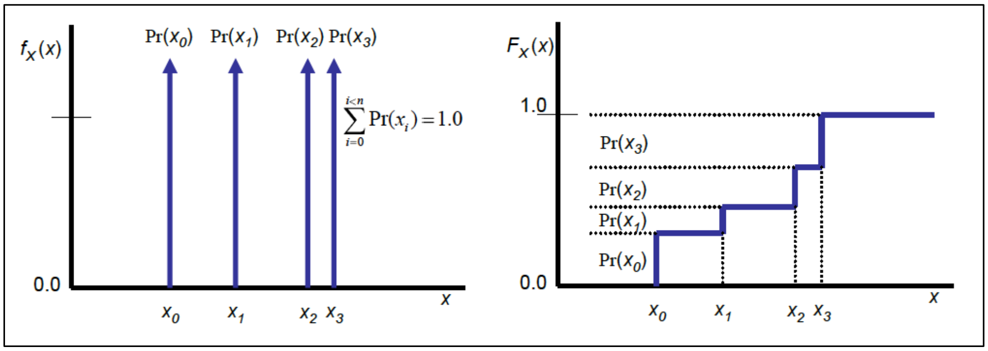

Figure 8: Discrete Distribution PDF and CDF [3]

From these distributions, estimation can take place to match a PDF and CDF to each individual system or component. With knowledge about how a system is expected to operate one could estimate the mean in the triangular distribution, for example, and include a region of distribution around this point. Proper selection of the distribution allows the ship designer to incorporate variation directly into the model, instead of relying on margins to deal with uncertainty.

Once each load has been assigned a distribution for each condition a simulation process must be completed to create a total overall loading profile. The method prescribed in DDS 310-1 as the most common method is the Monte Carlo simulation. In this method random variables represent an input to each component, assigning a load based on the distribution. The total load is then a summation of the loading of each component. This simulation is

then run for a large sample group to determine relevant output statistics. An example of the Monte Carlo simulation method is shown in Figure 9.

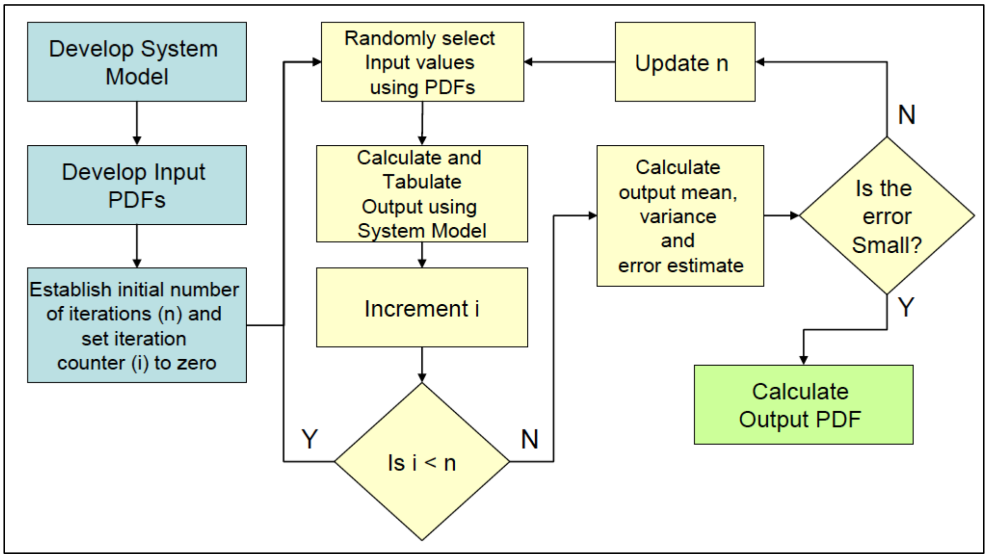

Figure 9: Monte Carlo Simulation Algorithm [3]

Using this method to quantify electrical loading conditions is appealing, once the distributions are estimated it is relatively straightforward to develop output statistics. The relationships between load and output are not based on first principles, therefore the method is computationally efficient and does not require solving equations of state.

Stochastic analysis provides more robust output results than the simpler load factor analysis. For each loading condition of the ship there will not only be an estimated mean value for the load, but a standard deviation. Understanding the range of expected loading provides improved information for sizing the electrical distribution. The EPLA produced through stochastic analysis still does not predict how the actual loading will occur over time, but provides better insight into the characteristics of the electrical plant.

1.4.3 Modeling and Simulation

Modeling and simulation is the most complex of the three methods for performing an EPLA, as it requires defining how components behave over time and interact with other components. As a result of the significant investment required to perform the modeling and simulation method, DDS 310-1 indicates this method is only invoked when specifically required. Examples presented of when this would be of use are when loads are large relative to generation capacity, when loads have abnormal characteristics, or when the loads cannot be modeled by the means discussed above [3].

One example of a load requiring modeling and simulation in today's Navy is the development of electromagnetic rail guns as a future naval weapon. When fired, these systems have large pulse loads that could have large transient effects on the electrical distribution system. In studies of future rail gun technology, the design has been focused on achieving a projectile muzzle velocity of 2500 m/s and kinetic energy of 64 MJ [12]. In studies surrounding this design, the thermal and electrical demand of such a system have been analyzed to determine the specific design issues presented to ship integration. An example electrical transient for a railgun from one of these studies is shown in Figure 10. In this example the rail gun has four track segments that fire sequentially to launch.

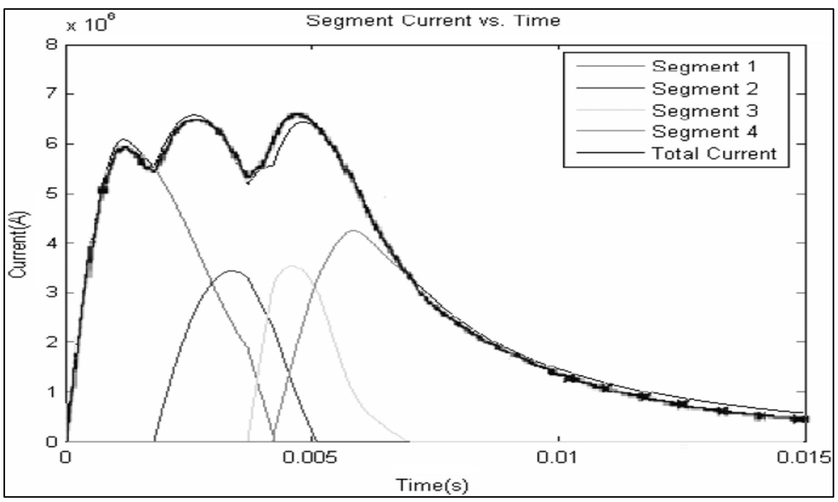

Figure 10: Rail Gun Electrical Transient [13]

In this example it is important to note the time scale associated with the model, which is on the order of milliseconds. Solving computationally intensive differential equations presents a means of accurately modeling the transient behavior seen as a result of firing the rail gun. This may be required to ensure proper powering and heat rejection for the weapon, but would not be practical as a means of determining the long-term behavior of total ship electrical load.

While modeling and simulation clearly presents the most accurate method of creating a representation of the loading of the electrical plant it also does not lend itself well to being used as an early-stage design tool. In the example of the rail gun the system is fully defined, allowing a model based on first principles to be developed. In many cases the specific loads may change as the design progresses, requiring flexibility on the part of the design tool.

1.5 Potential Improvements

As shipboard electrical power demands continue to rise, the need to accurately estimate future consumption will become increasingly important. Understanding how the fleet of today is operating and using this information to develop improved tools for future design work is critical. As an exemplar of a well-documented Navy shipboard power system, the DDG-51 will be used to evaluate current means of determining an EPLA. The following sections of the thesis will begin by discussing the sources of information available for the DDG-51 class and their applicability to this effort. Using these available resources, existing load factors will be updated to reflect current fleet operations. Finally, the framework for a behavioral model as an alternative to current modeling and simulation practices will be developed. The behavioral model combines the computational efficiency of stochastic methods with an understanding of ship behaviors to simulate electrical plant responses based to variations in input parameters. This modeling method would provide increased capability to predict ship behaviors over time, an important step towards meeting the power demand challenges presented in future years.

2 Sources of Information

Several disparate sources were brought together in this thesis, originating in an effort to update the speed-time profile for the DDG-51 class. Each of these sources will be discussed in greater depth in the following sections of this thesis.

Development of an updated DDG-51 class operating profile required raw operational data, which was gathered from visits to the fleet concentration areas of San Diego, CA, Norfolk, VA, and Pearl Harbor, HI. Individual DDG-51 class ship were visited on these trips, and engineering and deck logs were collected and manually collated to create a database of operational data. Using this database the amount of time spent at each speed was determined, creating the desired update to the speed-time profile. Additional mission or engineering mode profiles were also developed to assist in understanding the operational behaviors of the ship class.

In an attempt to gather operational data in an electronic form, data from the machinery control message acquisition system (MCMAS) was acquired from the Navy's Ships Systems Engineering Station (SSES). The MCMAS program contains a record of the configuration of the ship's systems over time, which provided a basis for understanding and profiling equipment behavior.

The DDG-51 program office also provided several crucial documents that aided in the development of the concepts seen later in this paper. These documents included the current references for the DDG-51 class, such as the EPLA and electric plant schematics, but also included a baseline report produced as part of the new construction process for the DDG-111 (USS SPRUANCE). The baseline report contains plots of the operating load for many components onboard the ship, which formed the basis for modeling the electrical profiles in the simulation portion. The original data used to create these plots was also provided, which allowed simple incorporation of the data desired.

2.1 Development of Updated DDG-51 Operating Profile

The speed-time profile for a ship class or type is the aggregate assumed utilization for the vessel over time. This relates a speed, or range of speeds, to the percent of time the ship is expected to operate in a given condition. In DDS 200-2 the operational profiles for various classes are given as a method to assist in calculating the annual energy cost to operate a ship. An example of the speed-time profile for the DDG-51 class is given in Figure 11.

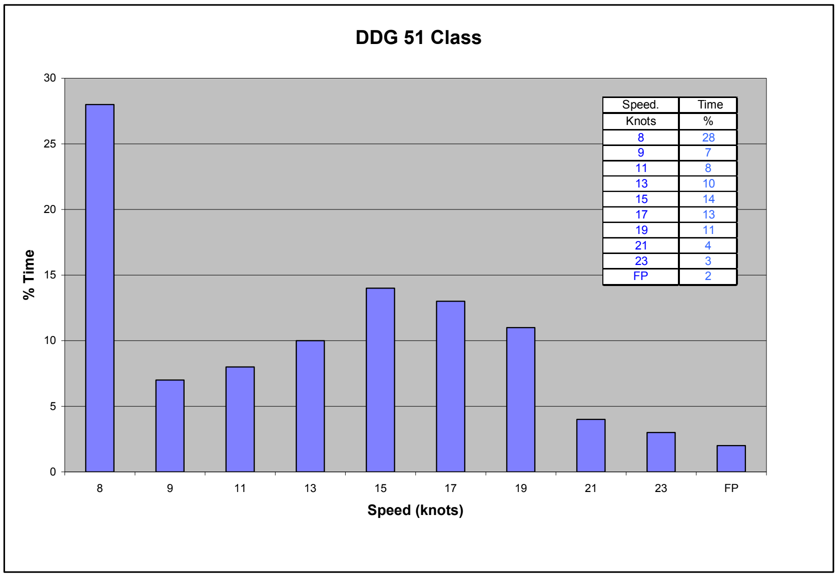

Figure 11: Current DDG-51 Class Speed-Time Profile [14]

This speed-time profile for the DDG-51 was based on an operating profile of the CG-47 class cruisers, developed in the early 1990's in a study performed by the Westinghouse MTD division [15]. It was assumed that the combatants with similar propulsion plants (two shafts, two GTMs per shaft) would be operated in a similar manner. Guidance within NAVSEA directed the application of this speed-time profile for use, with an assumed operating profile of the for the ship's engineering mode of 20% trail shaft, 60% split plant, and 20% trail shaft [16].

Development of a modern operating profile for the DDG-51 class was undertaken under the sponsorship of Naval Sea System Command's (NAVSEA) electric ships program office (PMS-320). The mission of the PMS-320 is to develop advanced propulsion and power distribution systems to the US Navy [17]. The collection and processing of the information provided in this thesis was conducted cooperatively with Travis Anderson and Katherine Gerhard, and was previously formally presented as a technical report to interested members of the NAVSEA community [18].

2.1.1 Data Collection and Database Development

The primary source for the development of a new operating profile was the ship's logs. The ship's logs are the legal record of the events occurring onboard the vessel, providing an accurate means of garnering the relevant information. The deck log and engineering log were collected from each DDG visited. The deck log provides an accurate reflection of the operations, speed, course, and rudder angle the ship took during a given day. The engineering log provides an account of the status of major engineering plant machinery status over the course of a day. Each of these logs then provided a different element required for the study, deck logs provided the speed and mission of the ship, while the engineering logs provided the configuration of the propulsion and electrical generation systems onboard.

To gather the required information visits were performed to the fleet homeports of San Diego, CA, Norfolk, VA, and Pearl Harbor, HI. The logs from a total of 16 DDGs were collected by manually scanning copies of the original logs stored onboard. A summary of the ships visited is shown in Table 3.

Table 3: DDGs Visited for Data Analysis

Hull	Name	Flight	Homeport	Hours Of Data
52	USS Barry	I	Norfolk, VA	2184
53	USS John Paul Jones	I	San Diego, CA	2208
59	USS Russell	I	Pearl Harbor, HI	2177
61	USS Ramage	I	Norfolk, VA	2206
70	USS Hopper	I	Pearl Harbor, HI	783
84	USS Bulkeley	IIA	Norfolk, VA	1968
87	USS Mason	IIA	Norfolk, VA	1488
88	USS Preble	IIA	San Diego, CA	2185
90	USS Chafee	IIA	Pearl Harbor, HI	192
91	USS Pinckney	IIA	San Diego, CA	2193
95	USS James E. Williams	IIA	Norfolk, VA	1438
97	USS Halsey	IIA	San Diego, CA	2159
100	USS Kidd	IIA	San Diego, CA	3418
102	USS Sampson	IIA	San Diego, CA	744
103	USS Truxtun	IIA	Norfolk, VA	1407
104	USS Sterett	IIA	San Diego, CA	1464

To ensure that the data was representative of actual ship operations a 3-month period of each vessel's data was collected when practical. A longer time period evaluating fewer ships was considered, but this approach was rejected since it could allow an outlying data set to have an undo impact on the final result. These logs represent the widest possible data available, including units with home ports in both the Altantic and Pacific, units operating both near home port and deployed, and units from different flights of the DDG-51 ship class. The time period for the data collected is recent, with logs dated from September 2011 to August 2012.

Each set of logs was compiled by manually entering the time of each relevant ship status change change in the engineering and deck logs into an excel spreadsheet. The spreadsheets for these ships were then transferred into MATLAB, where an overall database of operational data was generated. This database contained the times of each speed, mission, GTM, and GTG change across all ships included in the study.

2.1.2 Speed-Time Profile Results

The primary goal of this study was to develop an improved operating profile for the DDG-51 class ship. Using the database created from the various ship's logs the data was sorted to determine the amount of time spent, across all vessels, at each speed. Dividing the time at each speed by the total operational time, an updated speed-time profile for the entire ship class was created. This updated profile is shown in Figure 12.

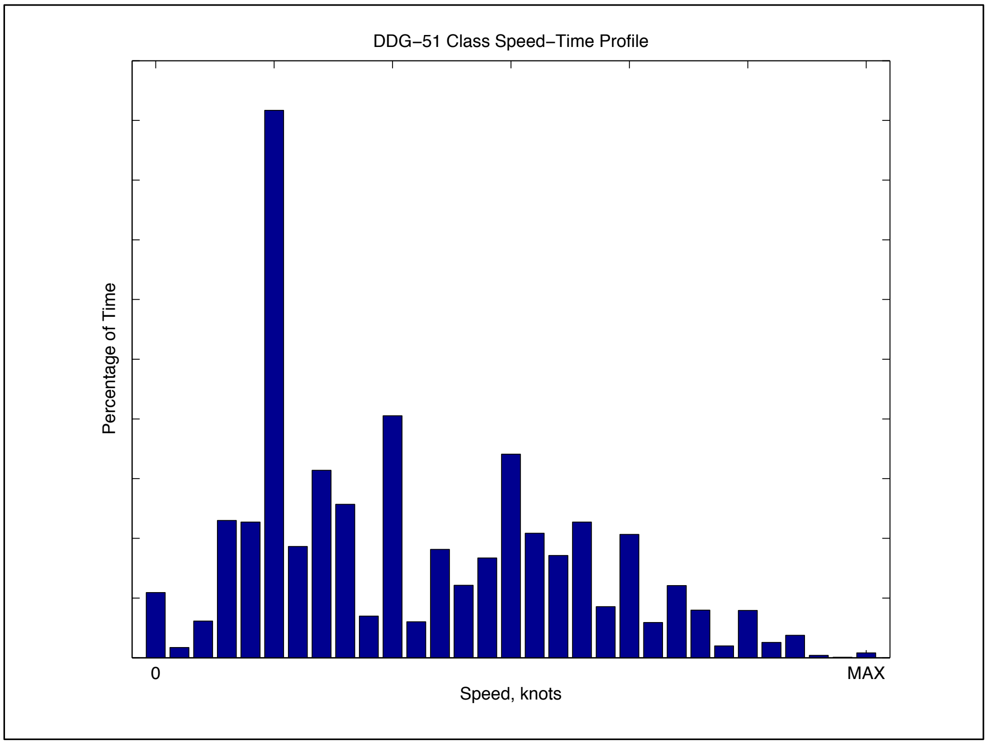

Figure 12: Updated DDG-51 Class Speed-Time Profile

Comparing the current DDG-51 class speed-time profile (previously shown in Figure 11) and this updated speed-time profile, a shift towards slower speed operations exists. This comparison can be best can be seen through the use of the cumulative distribution function (CDF) of each in Figure 13. This figure shows a clear shift in the operation of ships towards lower speeds, with roughly 55% of the operating hours of he DDG-51 class occurring at or

below 10 knots. The current profile assumes only 40% of operating hours below this point, indicating the updated profile demonstrates a dramatic shift towards slower operations.

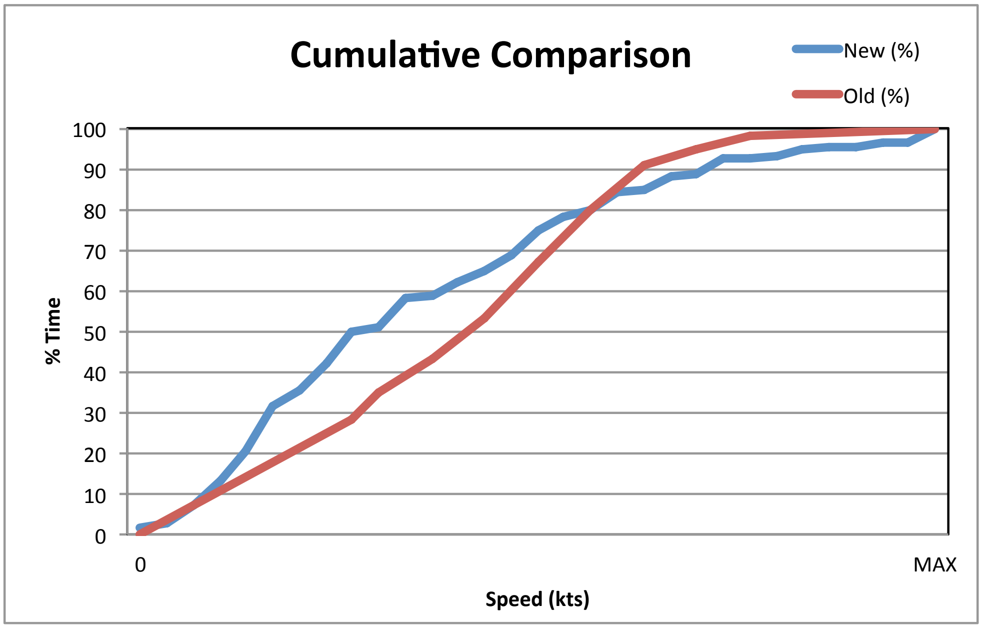

Figure 13: CDFs for New and Old Speed-Time Profiles

In addition to the speed of the ship, the database created for this study also included the time of each engineering plant configuration, ship mission, and GTG change. Utilizing this data, the ship's operations were evaluated based on the ship's engineering mode and mission. Performing this evaluation provides two separate pieces of information: how much time the ship spends in a given mission or engineering mode, and what the speed-time profile looks like while the ship is in this mission or engineering mode. The amount of time spent in each engineering mode and the associated speed-time profiles are shown in Figure 14.

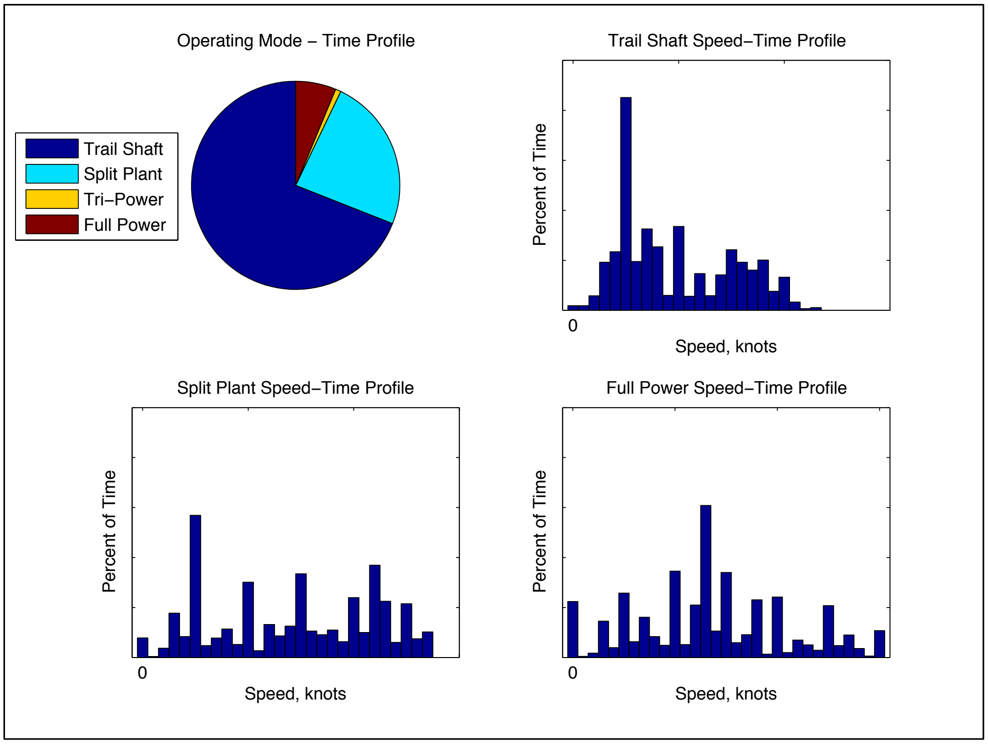

Figure 14: DDG Speed-Time Profile By Engineering Mode

Bart Sievenpiper 4/9/13 12:46 PM
Comment [1]: FIX these for 30+

Current NAVSEA design standards assume the ship operates 20% of its time in trail shaft mode, 60% of its time in split plant mode, and 20% of its time in full power mode. The data from current fleet operations demonstrates that this is clearly not the case, fleet operations are dominated by the use of the trail shaft engineering mode. This mode is seen in operation during nearly 70% of all fleet operations. This shift could have dramatic impacts on the assumptions that are used in the ship design process.

For each operating mode the profile exhibits strong shifts in the related speed-time profile. The full power mode is the only time that the highest percentage of operating speed does not occur at 5 knots, but rather at 13 knots. This is driven by the relationship that the ships is typically in full power only when in restricted maneuvering doctrine (RMD) condition. In this condition, the ship is placed in a maximum reliability due to the nature of the operating

environment of the ship. As a result, full power can be shown to be used less for achieving maximum possible ship speed, but rather for the reliability it provides.

Strong profile difference between different missions could also be expected. The three mission states developed in this study are RMD, transit, and operations. RMD is a period in which the ship is placed into a condition of maximum reliability, and these periods are clearly defined in ship's logs. The transit mission refers to periods during which the ship was moving between assigned locations. These periods are also well documented in ships logs, as each page of the deck log either records the operating area of the ship or where it is in passage to. The operations mission refers to the periods of time when the ship is performing functional duties as assigned, other than in transit or in RMD. While it may be desired to break this down into subordinate missions, the overlap of missions and variations in the way these are recorded in the deck log made this impractical. Plots for each mission can be seen in Figure 15.

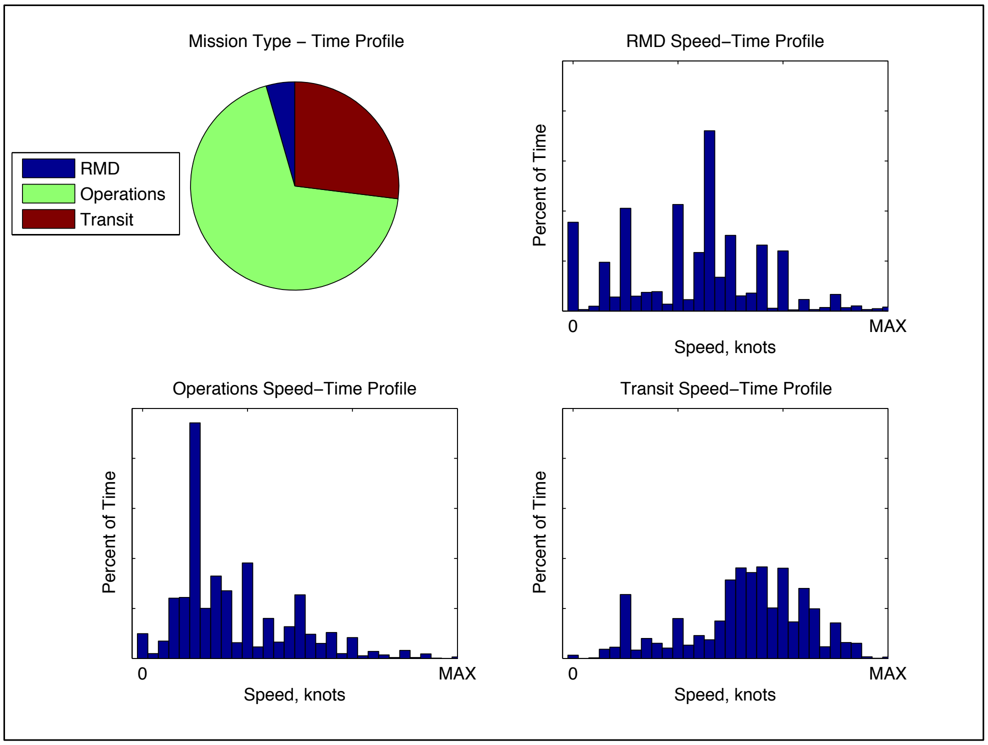

Figure 15: DDG Operating Profile by Mission

The comparison of the mission profiles provides several insights into the overall profile of the ship. The operations profile, by far the largest mission area, is the also the period driving low-speed operations. Since many operations involve station keeping at a given location, the low speed operations make sense. Transit operations requiring long distances, which drives a higher speed-time profile. The expected correlation between the RMD mission and full power operating mode is readily visible when comparing those profiles.

2.2 Machinery Control Message Acquisition System

The DDG-51 class was designed so that each vessel had a machinery control message acquisition system (MCMAS) onboard at the time of delivery [19]. This system was designed to provide a record of the state of equipment in the engineering plant over time, and provide a means of troubleshooting equipment in the event of plant casualties. In 1996 the integrated equipment assessment system (ICAS) was officially developed, and integrated into the existing MCMAS system [19]. The ICAS system provides the foundation of the condition-based assessment program in the US Navy, and considerable resources have been devoted to increasing the number of engineering components monitored.

The data collected onboard by the MCMAS system is stored in a centralized database, located on a dedicated computer terminal in the ship's central control station. The program includes a graphical interface that provides the option to analyze components monitored by the system as either a text file or as a graphical time series. Data is available in a variety of data rates, ranging from 1 to 300 seconds. Most engineering systems are monitored in the MCMAS system, including propulsion, electrical distribution, and auxiliary systems. The indications available depend on the system and can range from the simple on/off status of a pump to very detailed analyses such as discharge pressures or vibrations. Of particular interest for this thesis is the data recorded from the electrical distribution system, which monitors the positions of major breakers, as well as the current, voltage, and frequency of the power at those breaker locations.

In an effort to develop the DDG-51 operational profile using MCMAS data, several months of actual data from the USS HALSEY (DDG-97) was obtained from SSES. While this data was not ultimately fruitful in producing the speed-time profiles, it does provide a wealth of information for evaluation in this thesis. The months of data obtained correlate with the same time period for which the ship's engineering and deck logs were previously obtained. This correlation was vital for the interpretation of the data within the MCMAS system; without the ability to understand the functional configuration of the ship at a given time it is difficult to interpret a series of outputs from the engineering plant.

2.3 USS SPRUANCE (DDG-111) Baseline Report

As part of the construction process for the USS SPRUANCE (DDG-111), Alaris Companies, LLC, performed a study. The study performed a baseline power analysis for the ship during the performance of builder's trials. (The builder's trials are a period of time in which the construction yard takes the ship to sea, testing systems to ensure that thee vessel meets all required specifications.) During this time period a power reading was logged for hundreds of components on the ship, showing the steady state and/or transient behavior depending on the load.

The report and original data used to generate the report was provided by the DDG-51 program office to assist with the performance of the analysis performed in this thesis. The data in this report provides what is essentially the electrical fingerprint for a given load, exhibiting its electrical operating characteristics. This will be used later in this thesis when evaluating load behaviors independently of system behaviors.

In addition to the individual component power traces, the baseline report determined an average operating load, and average annual power consumption for the ship. This is useful in understanding how the power consumption actually seen for the DDG-111 compares to the data predicted in the EPLA for the DDG-51 class.

2.4 Non-Intrusive Load Monitoring

The use of Non-Intrusive Load Monitoring (NILM) has been investigated for many years in the Laboratory for Electromagnetic and Electronic Systems (LEES) at MIT. Several theses in recent years have explored the uses of power monitoring for potential uses ranging from supervisory control [20] to diagnostic monitoring [21] [22] [23] to monitoring aggregate shipboard load [24]. By monitoring the current in each phase and system voltage, the NILM device uses computational methods to determine the power signature for the monitored system. For more complex systems, such as power panels with multiple loads, the NILM uses frequency analysis to disaggregate composite loads into individual components [20].

To facilitate the testing and operational implementation of future technologies the Navy maintains a land-based engineering site (LBES) in Philadelphia, PA. This location has an operational model of the Main Engine Room #2 of a DDG-51 class engineering plant. The LBES includes the propulsion and electrical generating equipment, including GTM, GTG, a main reduction gear (MRG), and shafting to mimic the operational vessel [19]. In several previous academic efforts a NILM has been used at the LBES to demonstrate the ability to monitor loads and perform supervisory functions for the engineering plant [20] [25] [26]. In each thesis plant components were monitored with a NILM, and potential shipboard applications were developed. For the purposes of this thesis this provides a series of dynamic power traces with potential applicability in a modeling approach. For future generations of ships locations such as LBES could provide a means of recording power traces for developing power traces for predictive models before the ships are constructed. An example of the work performed in the thesis by Bennett in 2007 [20] is given in Figure 16. In this example, NILM power monitoring is being performed on 2A fuel oil service pump (FOSP) during the transition in turning on 2B FOSP and securing the 2A FOSP.

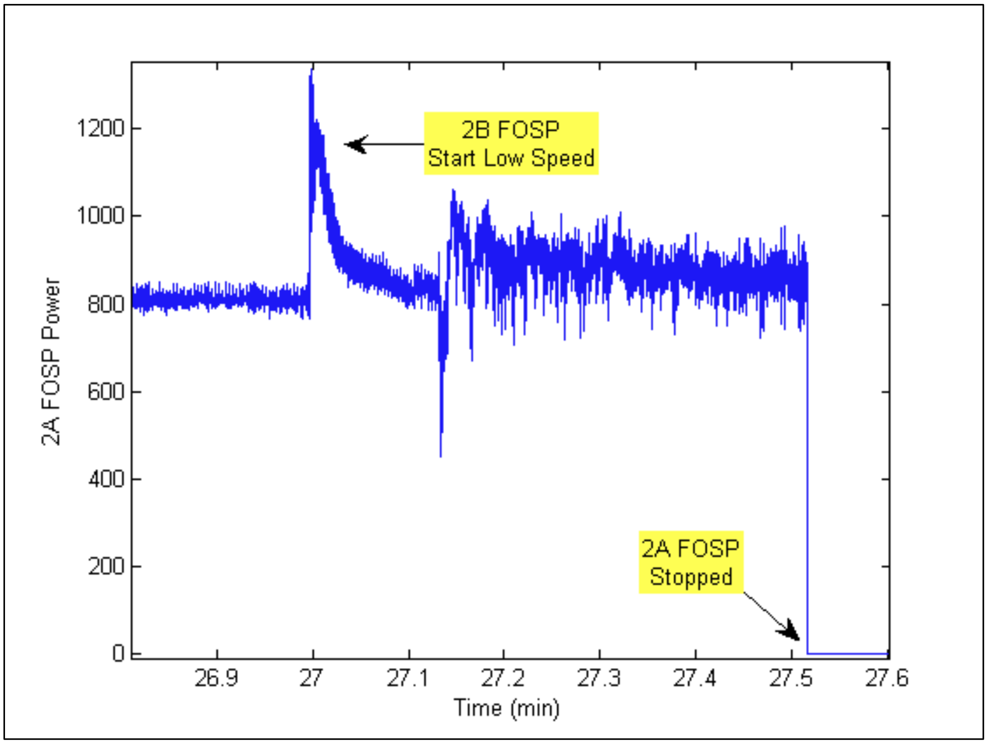

Figure 16: NILM Power Trace of LBES 2A FOSP [20]

By utilizing power traces from actual plant components at the LBES, another data set for the DDG-51 class for power transients over time becomes available for comparison. As shown in the above figure, many of these data sets include high fidelity images of plant transients, which could be used to develop system interrelations when developing system models.

3 Updating Existing Methods

Bringing together the diverse data shown section 2 of this thesis inspired investigation into its applicability to current design methods. The data has relevance in the design community today as the standard load factors in DDS 310-1 were not updated in its most recent revision [4]. Since the load factor analysis has historically been the means of producing an EPLA it is important that these values be as accurate as possible, reflecting the actual behavior in the fleet. Profiles for systems and their related components will be developed using both MCMAS data and information collected from the fleet operating profile.

Additionally, it is desirable to understand how easily power probability distributions could be constructed using the data developed above. Developing relevant power distributions is critical for the implementation of the stochastic load analysis methods of performing an EPLA.

3.1 Updating Load Factors

As introduced earlier, the load factor analysis method of performing the EPLA is the most straightforward method available. Using load factors for components directly from DDS 310-1 or developed based on expected system behavior, the power consumption in a given ship condition can be readily predicted. There is room for improvement within this process, however, as available fleet data is currently not being used to improve understanding of system behaviors. Through the use of MCMAS, an improved operating profile, and actual ship power traces improvements could be made in the assumed load factors for next-generation ship design.

It is important to recall that there are two pieces to a load factor: the load utilization and the power trace for the load. The load utilization of a component refers to the amount of time or operating state a component is operated, on average, in a given period of time. The power trace for a component is the actual power demand on the electric plant due to the operations for a given load over a given time. The manufacturer label plate data gives a rated load for a given motor or component, but many systems are constructed with motors

larger than that which is required to fully operate the system. The load factor is then the product of the utilization and the power consumption in operations, and understanding each side of this equation can yield process improvements.

Load utilization would be expected to be a more consistent parameter than the power trace of a component. Since system behaviors are often defined by standard operating procedures or operational demand, these values would be expected to differ less between different classes of ships. If the load utilization were known, the most conservative assumption where a power trace is not available would be to establish the load factor as equal to the load utilization.

3.1.2 Load Utilization Determination

By determining a load utilization profile for a component, it decouples system operations from specific component selection. While components in a system might change in a future ship design, understanding system behavior allows predicting load factors for future designs. By applying the expected power consumption to a known load utilization, a future load factor could be readily extrapolated for a next-generation system.

3.1.2.1 MCMAS Load Utilization

A starting point for examining how systems and loads are operated is through MCMAS. This program allows individual ship system and component states to be tracked over a time series, which can determine how loads are actually being used. As an example, a time series plot for the #2 lube oil purifier over the course of 23 days in February 2012 onboard a deployed DDG is shown in Figure 17. The lube oil purification system uses differential pressure from the reduction gear lube oil system to recirculate the operating fluid through a purifier. When running, the purifier is electrical powered to remove sediment through a centrifugal separator.

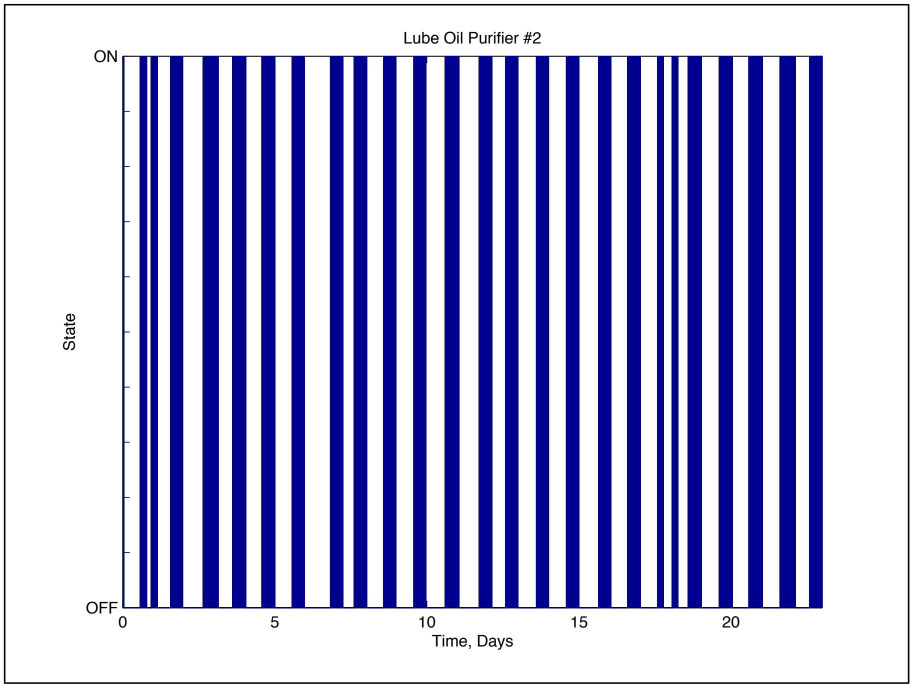

Figure 17: Operational Time Series for #2 Lube Oil Purifier

At the simplest level, the operations of the lube oil purifier utilization rate can be expressed simply as the fraction of time the purifier operates to the total time period. In the above demonstration, this would relate to a utilization rate of 0.43 for the lube oil purifier. Direct analysis of long periods of operating data also have the convenience that statistical periodicity of the operating cycle can be established, an idea that could be of further use for methods of stochastic load analysis or modeling and simulation. It is also important to note that the purifier has a subordinate component, the purifier lube oil heater, which operates when the purifier is in operation. This device operates thermostatically to control a desired temperature in the incoming oil. The lube oil heater would therefore also have the load utilization rate of 0.43, since it is in operation when the purifier is running.

Bart Sievenpiper 4/9/13 1:22 PM
Comment [2]: Are there more that I can put in here quickly?

The next system considered was the fuel oil transfer and purification system. This system allows for transfer of fuel between storage tanks and service tanks or the recirculation of a tank for purification. Plots for the transfer pump and purifier are shown in Figure 18.

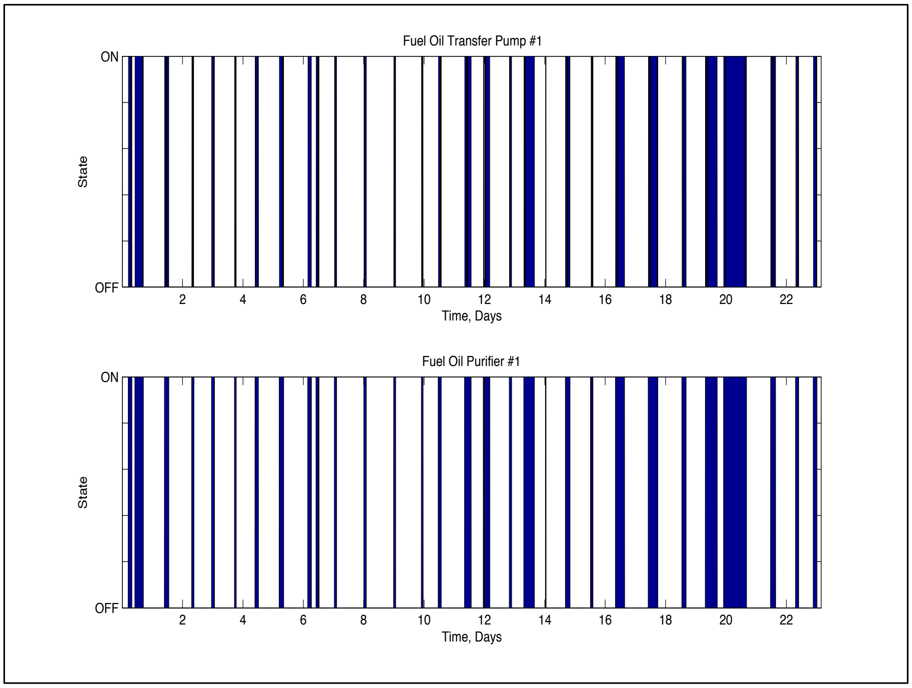

Figure 18: Fuel Oil Transfer Pump and Purifier Operation

It becomes immediately evident analyzing the operation of the system that these components are operated simultaneously; for each operation of the fuel transfer pump, a corresponding operation of the purifier occurs. In this case, these systems operate with a utilization rate of 0.23. Similar to the lube oil purifier, the fuel oil purifier heater is utilized in conjunction with the purifier and is assigned a utilization rate of 0.23.

Another system displaying a different type of cyclic operation is the potable water pumps. A potable water pump is typically in operation underway, providing sufficient pressure to provide supply throughout the ship. The system consists of two 100 gpm pumps, with normal operation consisting of one pump in operation and the other in standby. If demand causes system pressure to drop to a lower setpoint level, the standby pump will turn on to increase supply. Once system pressure reaches a higher setpoint level, the standby pump will secure. The recorded operation over time for these pumps is demonstrated in Figure 19.

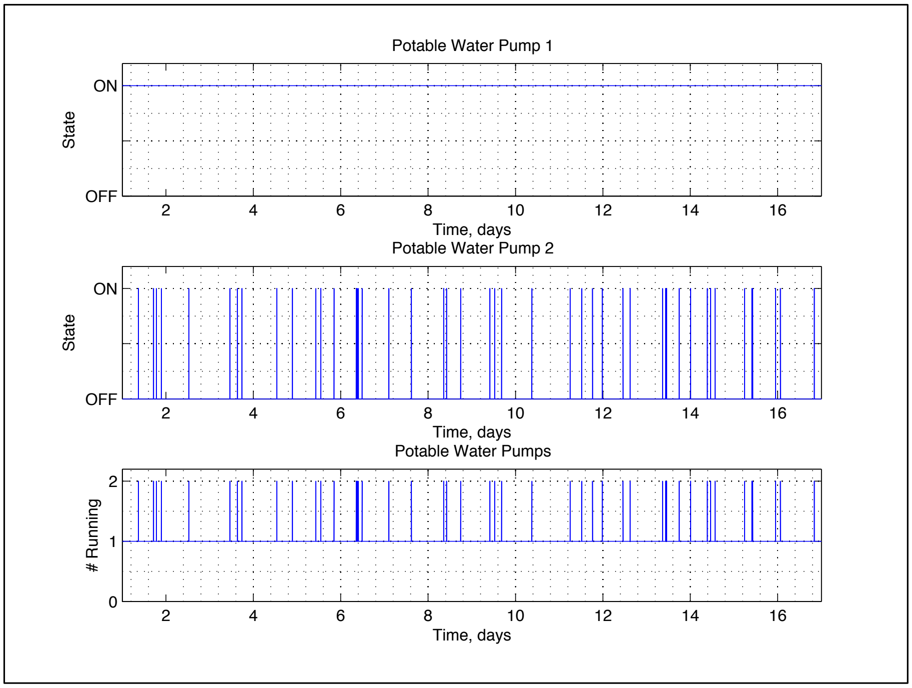

Figure 19: Operation of Potable Water System

In this system, the behavior exists as desired, with on pump running continuously, with the other pump coming on for short periods of time. Evaluating the running time, however, shows that the running period is minimal for the standby pump, in a 16 day period analyzed only 62 minutes of running operations for the standby pump was observed. This results in a utilization of 0.0027, which is essentially zero. The system overall then has a utilization of 1 for the first potable water pump and 0 for the standby pump.

The MCMAS system can also be used to validate the operating history of equipment systems. With many systems, such as fire main, there are multiple redundant pumps onboard the ship, with only a subset running at any given time. These systems tend to run for long courses of time without configuration change, a plot of the behavior they exhibit can be seen in Figure 20.

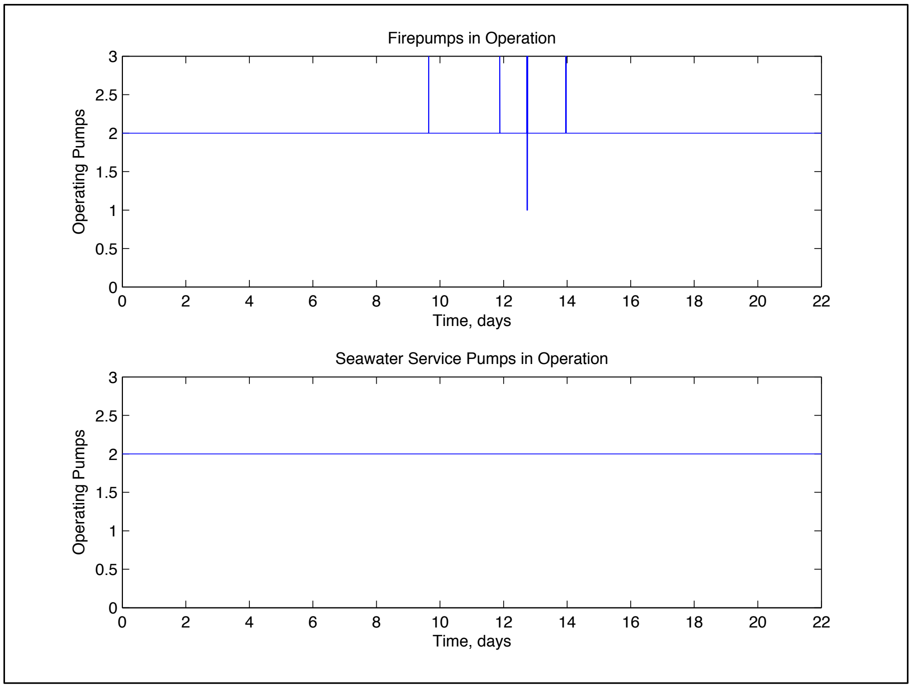

Figure 20: Fire and Seawater Service Pump Operating Profile

In this profile it can be seen that the expected condition onboard the ship at any given time would be two of six fire pumps and three of five sea water service pumps. Understanding the periodicity with which these systems reconfigure is relatively unimportant for the development of load factors, as the vast majority of the time a set number of pumps is in operation. For later development of behavioral modeling, however, understanding the manner by which the ship is operated becomes important. Temporary spikes can be seen in the operating number of running fire pumps, as a result of switching transients. A close up of the transient behavior exhibited by the fire pumps during day 12 of Figure 20 is shown in Figure 21.

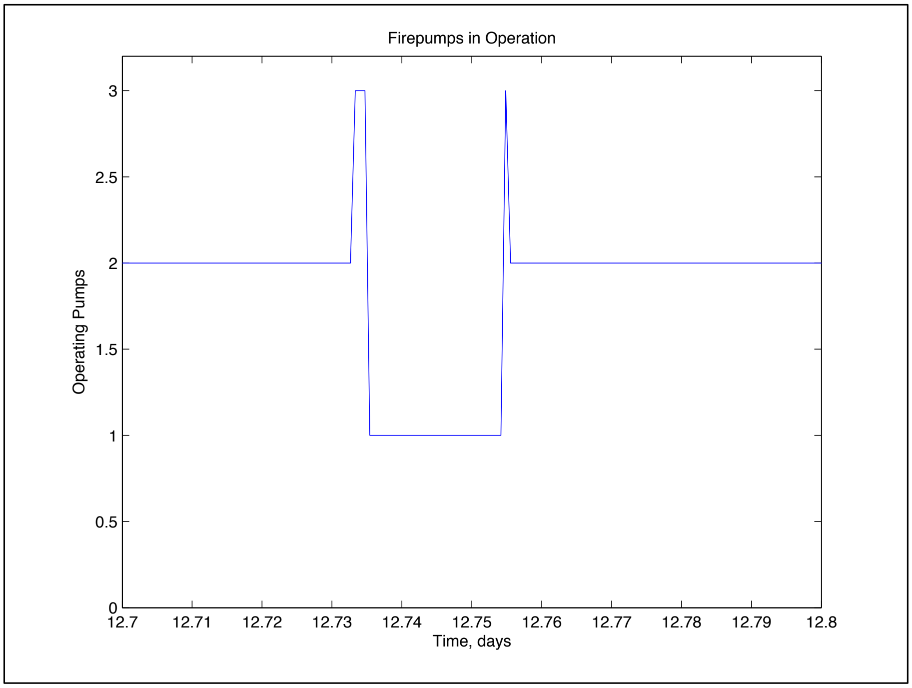

Figure 21: Detailed View of Fire Pump Switching Transient

The transient shown in the above picture demonstrates several characteristics; some of which are expected and some that are unexpected. For most fluid systems a pump switching evolution would bring an additional pump online prior to securing a running pump. It is unknown, however, what caused the drop to a single online pump in this figure. Infrequent conditions, such as equipment casualties or ship drill periods, could cause temporary configuration changes out of the ordinary. These minor deviations have little relevance when determining the long-term operating profile for a system.

The air conditioning plants, which are not pictured but are discussed in depth later in this thesis, operate three of five plants during summer cruise conditions. In some cases this can be used as validation for the operating state assumed in the EPLA, for the DDG-51 class the predicted value for fire pumps in normal cruising condition is one of six in operation. Validation is important not just for improving current load analysis, but also to provide feedback to ship designers about how the fleet is utilizing the systems.

A summary of the load utilizations determined using the MCMAS program is shown in Table 4.

Table 4: MCMAS Derived Load Utilization Rates

Component	Load Utilization Rate
Lube Oil Purifier (#1&2)	0.43
Lube Oil Purifier Heater (#1&2)	0.43
Fuel Oil Transfer Pump (#1&2)	0.23
Fuel Oil Purifier (#1&2)	0.23
Fuel Oil Purifier Heater (#1&2)	0.23
Potable Water Pump 1	1.00
Potable Water Pump 2	0.00
Fire Pump 1	1.00
Fire Pump 2	0.00
Fire Pump 3	0.00
Fire Pump 4	1.00
Fire Pump 5	0.00
Fire Pump 6	0.00
Sea Water Service Pump 1	1.00
Sea Water Service Pump 2	0.00
Sea Water Service Pump 3	1.00
Sea Water Service Pump 4	0.00
Sea Water Service Pump 5	1.00
AC Compressor 1	1.00
AC Chill Water Pump 1	1.00
AC Compressor 2	0.00
AC Chill Water Pump 2	0.00
AC Compressor 3	1.00
AC Chill Water Pump 3	1.00
AC Compressor 4	0.00
AC Chill Water Pump 4	0.00
AC Compressor 5	1.00
AC Chill Water Pump 5	1.00

The data in this table shows load utilizations using the data collected from a single vessel, and a more diverse data set would provide more accurate results. This data does, however, demonstrate the relative ease that simple data mining tools could turn unused fleet information into results that would be pertinent to the design community.

3.1.2.2 Fleet Profile Load Utilization

A second approach to determining the load utilization rate of a component is by considering its contribution to a system with a defined ship profile. Through knowledge of the engineering plant, load utilization factors could be determined through the DDG-51 class operating profile level discussed in section 2.1. This method has the added benefit of using a large and proven data taken across a large subset of the fleet, though a somewhat more limited applicability.

Guidance for the development of load factors directs that standby or redundant equipment is zero unless the equipment is actually running concurrently [3]. As a result, the loading will be considered as distributed evenly over the 1A and 2A engines with a contribution from the 1B and 2B engines only when the ship is in a full power operational mode. While this is not reflective of the actual plant operations, it ensures that for the long-term average loading for that portion (forward or aft main machinery room) is correct. These values can be seen as calculated in Table 5.

Table 5: Operating Mode Determination of GTM Utilization

Operating Mode	% of Time in Mode	1A	1B	2A	2B
Full Power	6	0.06	0.06	0.06	0.06
Split Plant	24	0.24	0	0.24	0
Trail Shaft	70	0.35	0	0.35	0
TOTAL		0.65	0.06	0.65	0.06

This indicates that in each plant over a long time history there will be approximately 65 hours where at least one GTM is running, and 6 hours when both GTM are running. Both of these values will become important when the one studies the inter-relation of the operations of the GTM and the equipment that operates in conjunction with it. One could simply roll the operating time of the 1B and 2B engines into the values for 1A and 2A (making each a 0.71 rate), however this would ignore certain dependencies that exist within the data.

3.1.2.2.1 GTM Services

Each GTM has two major direct support components that it depends on to operate properly, the cooling fan and enclosure heater. The first is the GTM cooling fan, which is a relatively large (130 HP) fan that promote exhaust gas removal and provide enclosure cooling. This fan will, therefore, be running whenever the GTM is in operation. Each GTM enclosure also has a small heating unit (8 kW) that operates to maintain temperature in the enclosure high enough to ensure suitable fuel viscosity for GTM starting. This thermostatically controlled heater is in operation when the unit is shutdown [27].

3.1.2.2.2 Fuel Service Pumps

Ensuring that all operating turbines (GTM and GTG) have sufficient fuel supply, the fuel oil service pumps are also required for GTM operation. Each plant has two fuel service pumps; one of which supplies fuel pressure and the other acting as a redundant pump. These are 2-speed positive displacement pumps, with a single pump capable of supplying two fully loaded GTG and two fully loaded GTM [28]. Using fuel consumption curves at each speed in each propulsion mode, the speed of the service pump at each speed can be calculated. To err conservatively, it is assumed that the pump is also supplying a fully loaded GTG. The fuel consumption curve for a single plant operating is seen in Figure 22. In this it is important to note that the trail shaft condition assumes one is monitoring the operating plant, at a given speed the operating shaft utilizes more fuel, but only one shaft is in operation.

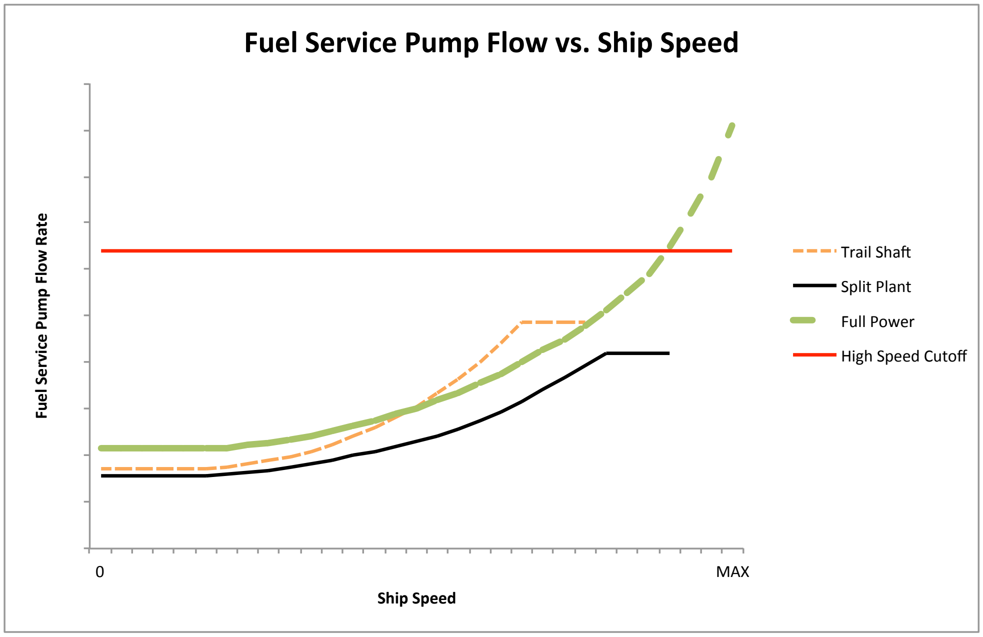

Figure 22: Fuel Service Pump Flow vs. Ship Speed

From these curves, it can be easily seen that very rarely is the ship operating at a speed and propulsion mode that would require a service pump in high speed. The ship spends less approximately 6 percent of its operational hours in full power, and only than 5 percent of full power time above the speed needed to reach the high-speed fuel service pump. As a result, the approximately 0.3% of the time the ship should reach this level becomes insignificant, and the EPLA should consider the pump only in the slow speed. Since it can be assumed that a GTG will typically be in operation in each plant and a positive displacement pump in half speed operates at approximately half power, the utilization rate of the 1A and 2A fuel oil service pumps should be 0.5. Since they will not run in this scenario, 1B and 2B fuel oil service pumps have a utilization rate of 0.

3.1.2.2.3 Main Lube Oil Pumps

To transfer the high-speed output of one or both operating GTM in a propulsion plant to the relatively low speed motion of the shaft, gas turbine powered ships utilize main reduction gears (MRG). The MRG require lubricating oil to prevent damage when rotating, which is

supplied from the main lube oil pumps to the main lube oil system. There are two two-speed electric pumps, one standby and one automatic backup, for each MRG. There is also a gear-driven pump that is mechanically coupled to the MRG that supplies lube oil flow to the system. In normal operation one pump (1A or 1B in one MER, 2A or 2B in two MER) would operate in slow speed. If system pressure drops below specification, the pump would ship into high speed. If shaft rotational speed is great enough, the gear-driven pump can provide sufficient pressure, and the electric pump will secure until speed drops.

Using fleet data for the shaft rpm compared to ship speed a plot can be developed to demonstrate the regions in which the electric main lubricating oil (MLO) pump will secure. This cutout speed is shown in Figure 23.

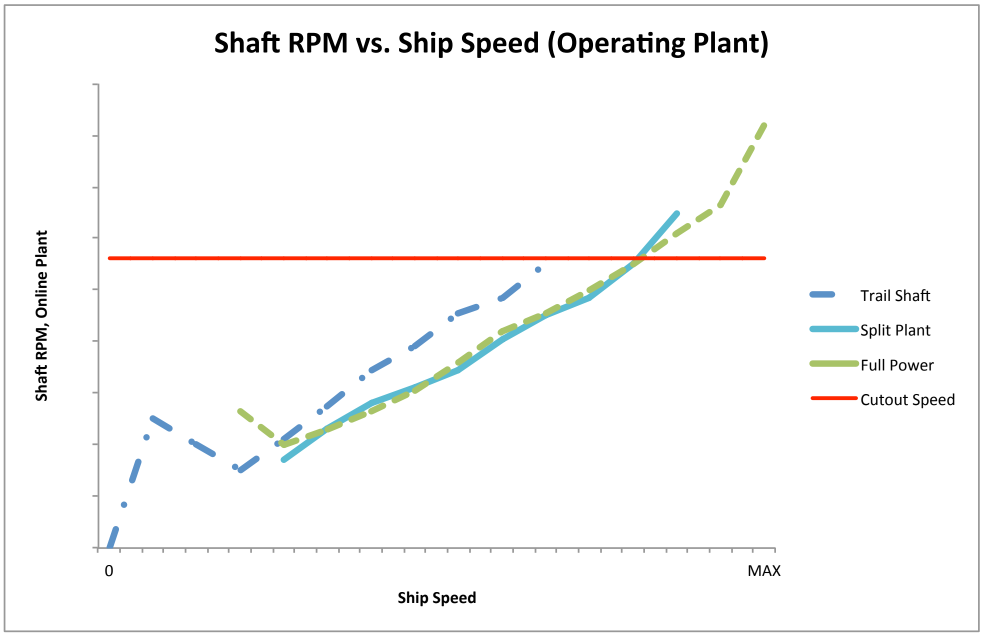

Figure 23: Shaft RPM vs. Ship Operating Speed

From this graph it can be seen that the electric pump will secure only in he highest ranges of ship speed in a given engineering operating condition. For the trail shaft condition, the operating speed never reaches the required shaft RPM in the operating plant, and the propeller spinning free on the other shaft will obviously not reach this limit. As a result, a

MLO pump in each plant will be operating. In the full power condition, the speed will be high enough for the electric pump to cutout 11.2% of operating time, and for the full power condition this will occur 11.9% of operating time. Using the previously established values for the operating time in each engine configuration (shown in Figure 14), then the electric MLO pump would be expected to be operating 96% of the time.

The final piece to defining this load factor is that the electric MLO pump is a 2-speed screw-type pump. The high-speed state is twice that of the low-speed state, and for this style of pump the electrical power consumed for the low-speed state is half that of the high-speed state. Since the pump in normal operating conditions would only operate in low speed, this means it would be operating at half power. Combining this with the amount of time the pump is expected to run, the load utilization for the electric MLO pump is 0.48.

3.1.2.2.4 Controllable Pitch Hydraulic Pump

There is one controllable pitch propeller (CPP) attached to each shaft, and each propeller has an independent hydraulic system to adjust the pitch of propeller blades. Similar to the main lube oil system, there is one pump driven mechanically by the MRG, and one electric pump. The electric pump maintains hydraulic system pressure when ship speed is low, and will automatically secure when the speed increases above a threshold value. The speeds associated with this transition are the same as those for the MLO pumps, and therefore the same operating rates can be assumed. Since the CPP electric pump is a single speed pump, this indicates an operational utilization rate of 0.96 for the pump in each engineering plant.

3.1.2.2.5 GTG Services

Typical ship operations for the DDG-51 include the use of 2 GTG to provide electrical power to the ship. Analyzing the DDG-51 class data set for at-sea operations, it is evident that this is a nearly exclusive operating condition for the ship class. This can be seen graphically in Figure 24. The region of single generator operations primarily occurred when maintenance on a running generator was required while another generator was down due to casualty. The region of 3 generator operations, or full power, was a transient condition during generator switching operations.

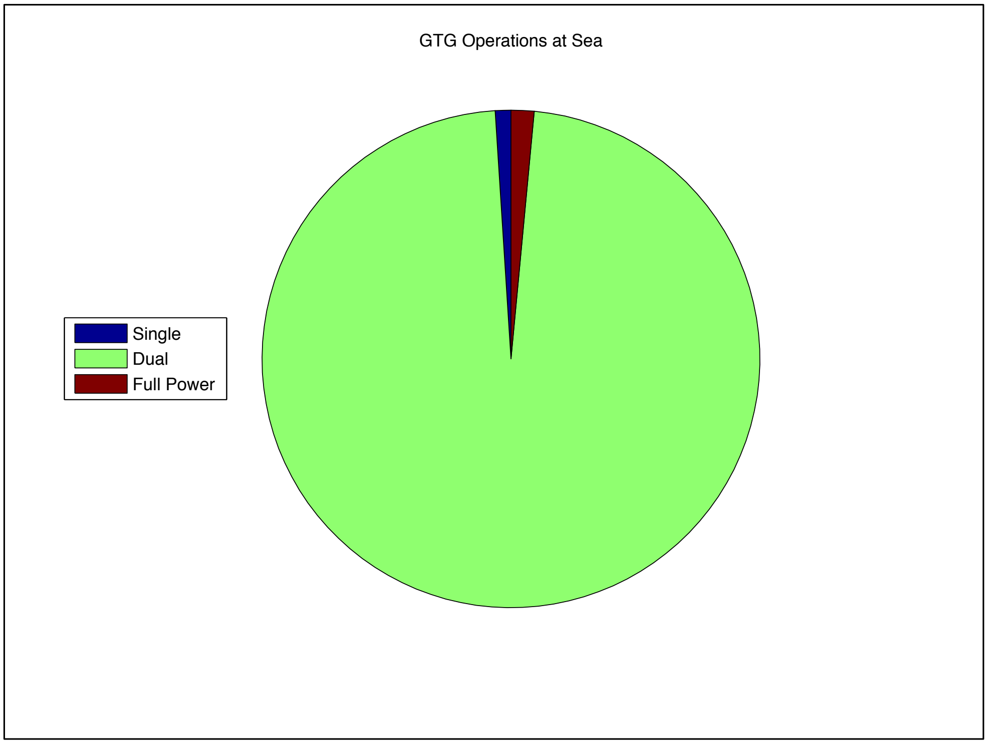

Figure 24: GTG At-Sea Operations

For the generator services required by a GTG the fuel service pump was considered in conjunction with the GTM, and will not be considered here. The other required services for GTG operation are the cooling fan, cooling water pump, and enclosure heater. For the GTG cooling fan and cooling water pump the utilization rate for operating generators should be 1.0, while the heater will have a utilization rate of 1.0 for the secured generator.

3.1.2.2.6 Load Utilization Summary

The overall load utilization determined using the fleet data discussed above is presented in Table 6. It is important to note that when determining the load factor, which is the product of utilization and the fraction of rated power consumed when operating, that the state of the equipment assumed below is understood. For example, it was assumed the MLO electric pump is operating in slow speed, so the fraction of rated power used to get the load factor should be that observed in slow speed operation.

Table 6: Summary of Load Utilization for Selected Propulsion Components

Component	Load Utilization Rate
GTM 1A Cooling Fan	0.65
GTM 1B Cooling Fan	0.06
GTM 2A Cooling Fan	0.65
GTM 2B Cooling Fan	0.06
GTM 1A Enclosure Heater	0.35
GTM 1B Enclosure Heater	0.94
GTM 2A Enclosure Heater	0.35
GTM 2B Enclosure Heater	0.94
GTG 1 Cooling Fan	1.00
GTG 2 Cooling Fan	0.00
GTG 3 Cooling Fan	1.00
GTG 1 SW Cooling Pump	1.00
GTG 2 SW Cooling Pump	0.00
GTG 3 SW Cooling Pump	1.00
GTG 1 Enclosure Heater	0.00
GTG 2 Enclosure Heater	1.00
GTG 3 Enclosure Heater	0.00
1A MLO Pump	0.48
1B MLO Pump	0.00
2A MLO Pump	0.48
2B MLO Pump	0.00
CPP Hydraulic Pump 1	0.96
CPP Hydraulic Pump 2	0.96
1A Fuel Oil Service Pump	0.50
1B Fuel Oil Service Pump	0.00
2A Fuel Oil Service Pump	0.50
2B Fuel Oil Service Pump	0.00

3.1.3 Power Requirements

The second half of determining the load factor for a system is defining the power consumption of the component within a system. One important reason to separate the power consumption from the utilization is that the design of the system and selection of components will affect the percent of rated load that a component operates at.

To help illustrate the differences that can arise, Figure 25 depicts the starting transients for the cooling fans from a GTG and GTM. These plots were created using the transient data

from the USS SPRUANCE baseline report, then normalizing the data with the EPLA rated load for each fan.

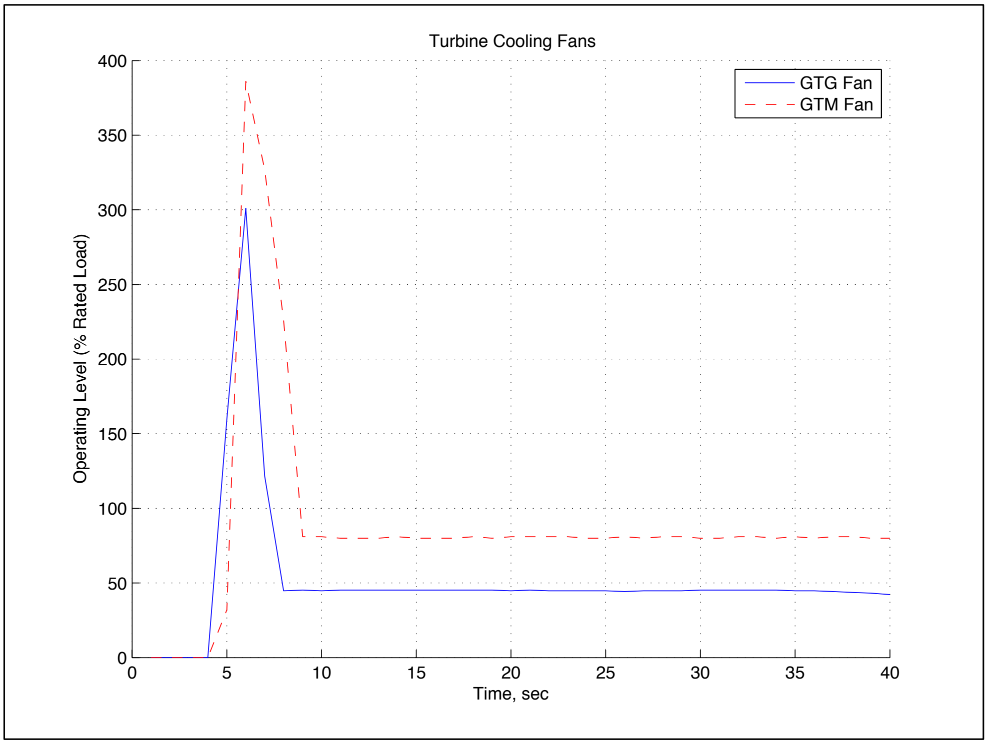

Figure 25: Comparison of Turbine Cooling Fans

It is readily apparent from this plot that there are strong operational differences between the GTG and GTM cooling fan. Following the starting transient, the GTM cooling fan runs at approximately 80% of its rated load, while the GTG cooling fan runs at just under 50% of the rated load.

Many reasons may exist for why there is a discrepancy in operating level between the different types of fans, but this issue presents a clear problem in determining the load factor. Since a component load factor is the product of its utilization rate and its average operating power, the final load factor should change depending on the system design and component selection. If a fan is oversized relative to the requirements of the system, it may

operate at a lower percent of rated load but have other selection properties desirable to the ship designer. Additionally, since each GTG has a different intake and exhaust arrangement, it is possible that each GTG cooling fan could have a different steady state operating point.

3.1.4 Load Factor Calculation

As previously described, the overall load factor is a product of the load utilization and power trace. To demonstrate the calculation of a load factor the lube oil purification system will be examined, which provides a system with two components that exhibit vastly different operating profiles; the lube oil purifier and the purifier heart. The lube oil purifier itself is powered by an induction motor, which spins the purifier. The operation of the purifier is demonstrated in Figure 26.

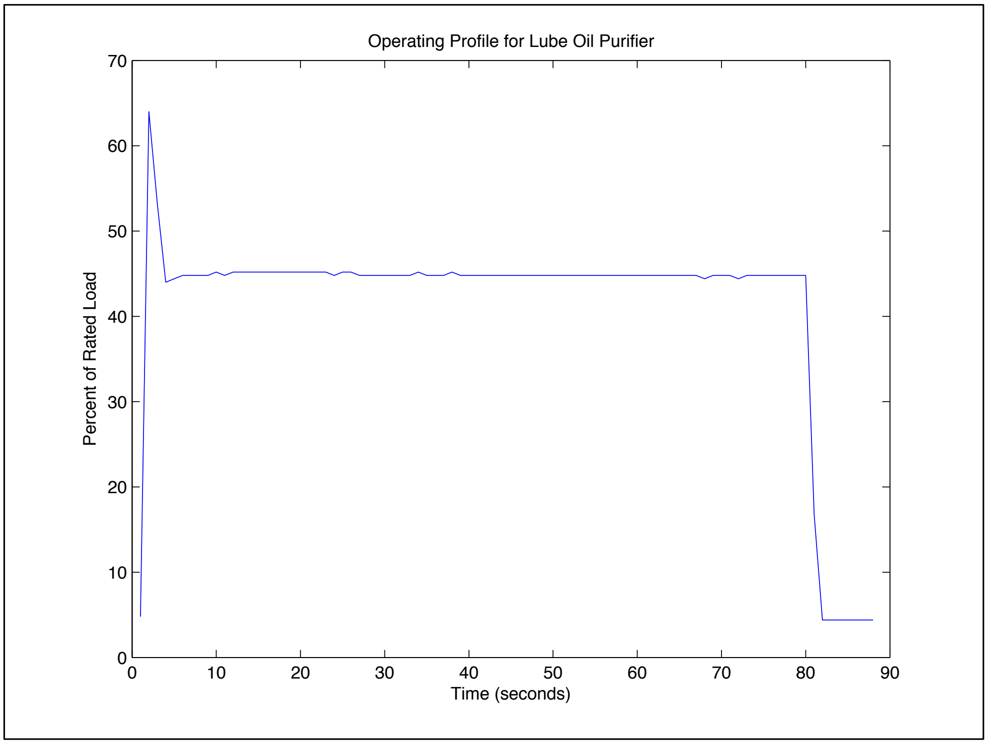

Figure 26: Operation of Lube Oil Purifier

The purifier demonstrates similar operation as the turbine cooling fans seen in Figure 25. The average steady-state operation of the purifier is 44.9% during the analyzed period. The purifier heater, however, demonstrates a much different profile, seen in Figure 27.

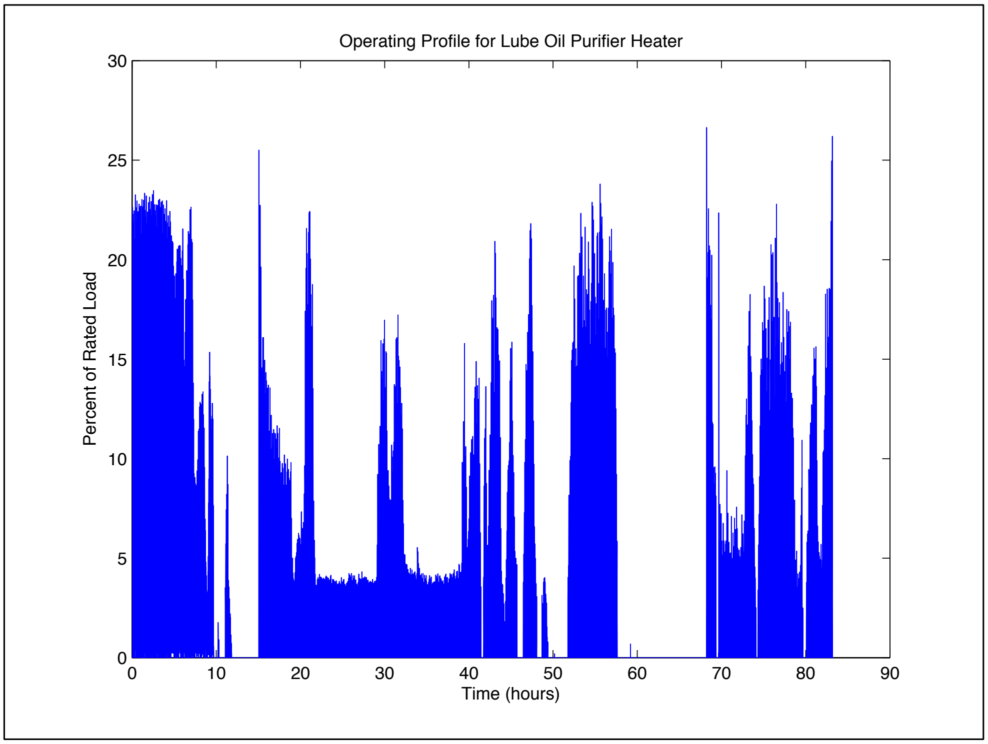

Figure 27: Operation of Purifier Heater

The purifier heater operates using a thermostatic controller to heat the incoming oil to the purifier to a certain level. The above profile shows the average operating level, recorded as averaged 30-second increments over several days of operations. The periods of inactivity seen in the graph correlate to periods when the purifier is not in operation. Taking the average operating level of the non-zero elements of the purifier heater's operations, the heater operates at 6.45% operating power.

Combining the information from the power profiles and combining it with the previously determined load utilization for the system a load factor can be determined. These results are shown in Table 1.

Table 7: Lube Oil Purification Load Factor

Load	Load Utilization	Average Operational Rate	Calculated Load Factor
Lube Oil Purifier	0.43	44.90%	0.19
Lube Oil Purifier Heater	0.43	6.45%	0.03

The result of 0.19 for the lube oil purifier seems relatively low, but the standard (not ship specific) value given in DDS 310-1 for the purifier is 0.3 [3]. This indicates that while the value has changed, it is not radically different. The load factor for the purifier heater is not specified in the DDS 310-1, but the value of 0.03 seems lower than may be expected. Understanding these differences could be an invaluable portion of post-construction validation of the EPLA in future ship classes. By validating the EPLA the sources of deviation from predicted values could be determined, whether it is from the operating rate or in the load utilization.

3.2 Stochastic Load Analysis

The inclusion of stochastic methods into DDS 310-1 occurred in the most recent revision of the design guide. Since it has not been a method used historically to conduct the EPLA, distribution functions are not readily available for shipboard components. Examples of methods to create a PDF for a given load using available data are presented in the following sections.

3.2.1 Developing a Distribution: AC Compressor Motor

Examining the data available within the MCMAS database the air conditioning (AC) compressor stood out, as a result of the system interface between the MCMAS system the AC plant. The AC plants onboard the DDG-51 operates by refrigerating a chill water loop and rejecting heat to seawater. The chill water is then piped throughout the ship to provide air conditioning and electronic equipment cooling. There are 5 AC plants onboard the DDG-51 class, labeled 1, 1A, 2, 3, and 4. MCMAS receives a direct reading of motor current, tonnage, and other system parameters at each time step, providing a direct means

of determining the distribution of motor current over time. Making an assumption for the approximate power factor for the AC plant, this motor current could be transformed into a kW loading for comparison to the EPLA.

Examining several weeks of data from MCMAS, the data was sorted using MATLAB for each AC plant. A distribution over this time period is shown in Figure 28 for each AC unit.

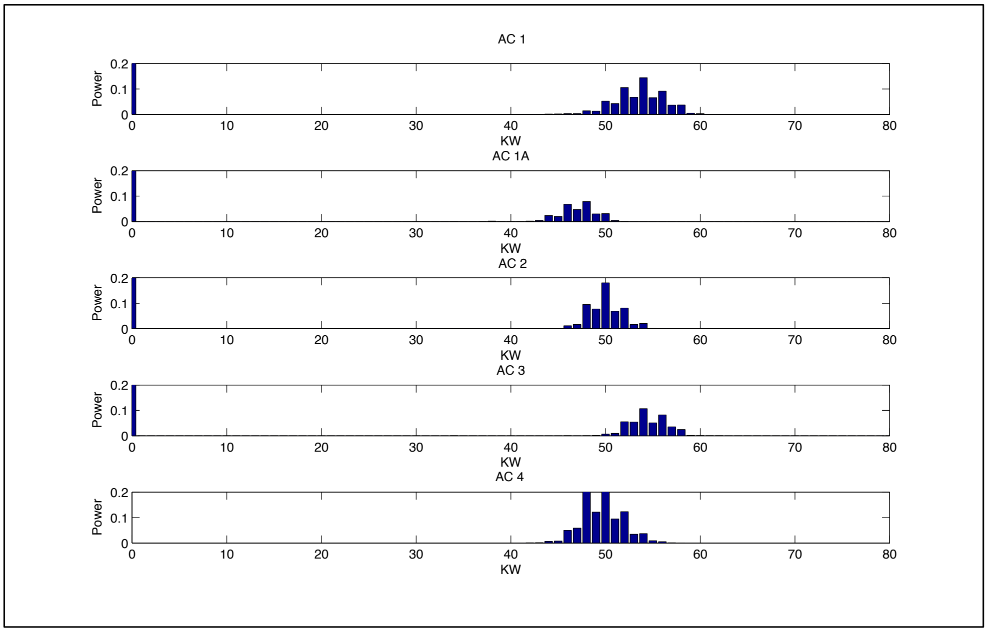

Figure 28: Individual AC Compressor Power Distributions

It is evident there is some differences in the mean between AC plants, but each plant demonstrates a similar pattern of loading. Differences in mean are not unexpected since there may be small differences in the chill water utilization between regions of the ship. Taking an average of the plants over the time period a PDF for the operation of a single AC plant was developed, as shown in Figure 29.

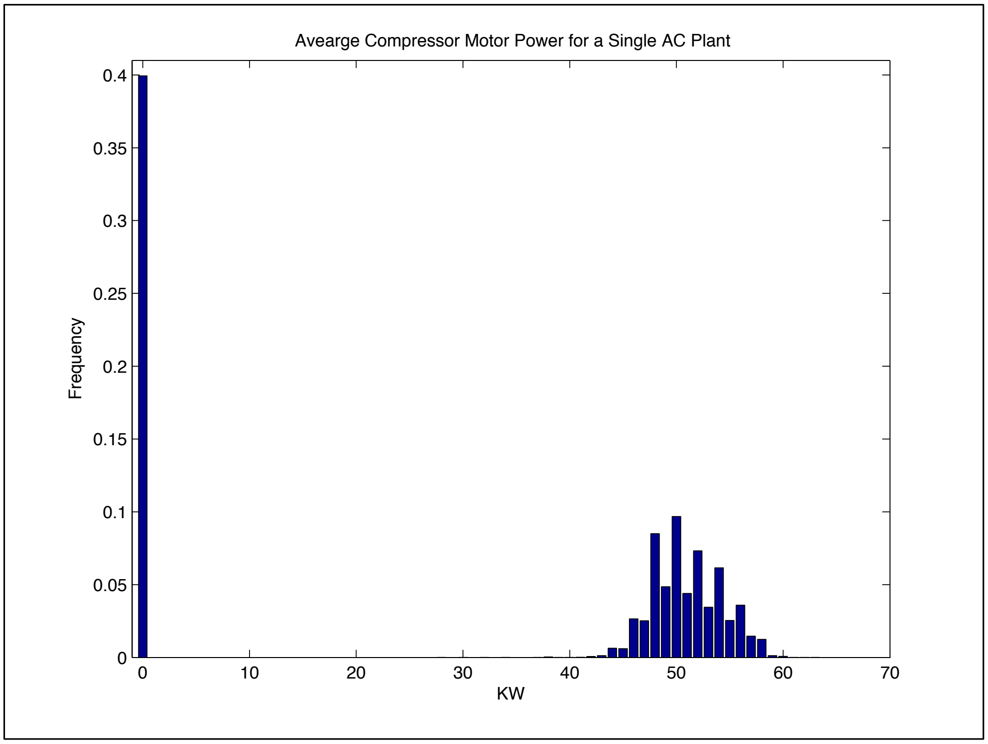

Figure 29: AC Compressor PDF

This PDF inherently contains two separate pieces of design information required by DDS 310-1, the plant configuration and operating distribution. The plant configuration can be inferred by the amount of time spent with no power, which occurs approximately 40% of the time. In the normal operating configuration 3 of 5 AC plants are operating onboard the ship, so this corroborates expectations. The other piece of information available from this PDF is the running distribution of power. With this known distribution, a triangular or normal distribution could be fit to the data set for the purpose of stochastic modeling.

3.2.2 Stochastic Modeling of Entire System

The PDF demonstrated in Figure 29 shows a relatively straightforward means of developing a distribution. It would be expected that using this PDF to perform a stochastic analysis, the average total power consumption among all AC plants would appear similar to the actual distribution seen in Figure 30.

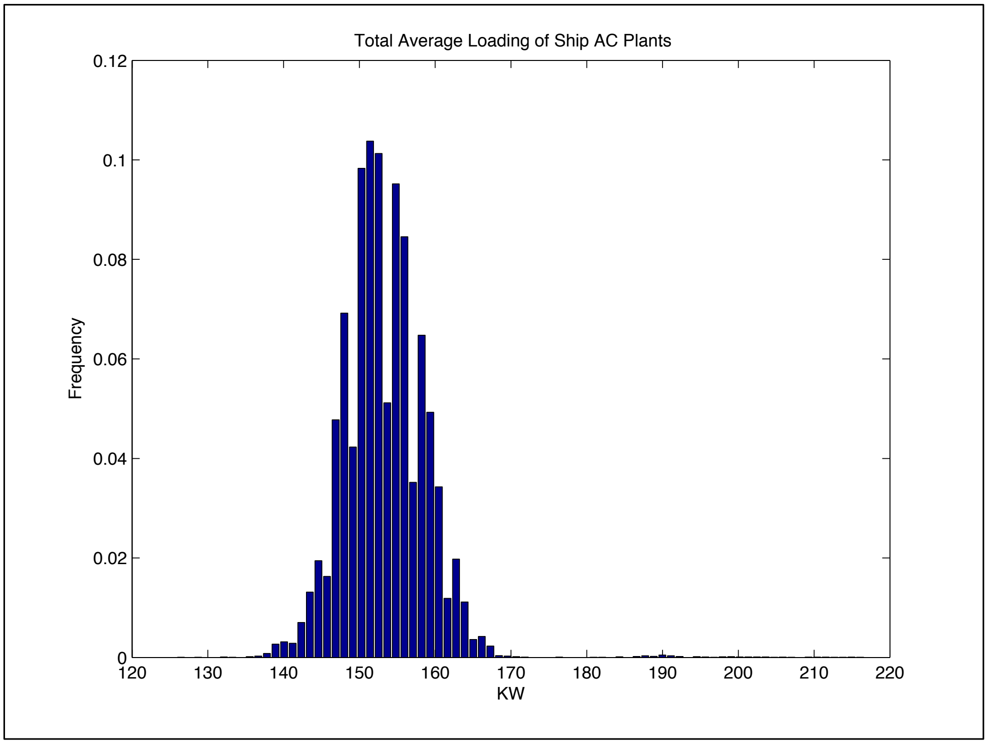

Figure 30: Average Cumulative Loading of all AC Plants

It is important to note in this figure the long tail to the right hand side of the graph (loading seen at approximately 190 kW) that does not exist in Figure 29 for the average AC plant. This tail is representative of the overlap time that exists when switching between AC plants, yielding a temporary condition where 4 AC plants are in operation. For a stochastic analysis of the entire ship to capture the peak loading conditions correctly seemingly minor details, like infrequent periods operating 4 AC plants, must be included in the stochastic model. A means of correcting this is type of problem is suggested in DDS 310-1, using a two-level model. In this method, the first level would be a variable that would define how many AC plants were running (3 or 4) with specific probabilities associated for each. In the second level a random variable generated for each operating compressor would generate a simulated load.

Implementing the stochastic method should be able to reproduce distributions that are similar to Figure 30, but an examination of the underlying AC plant data shows the method does not fully capture all effects. A plot of the total load consumed by all AC compressors is shown as a time series in Figure 31.

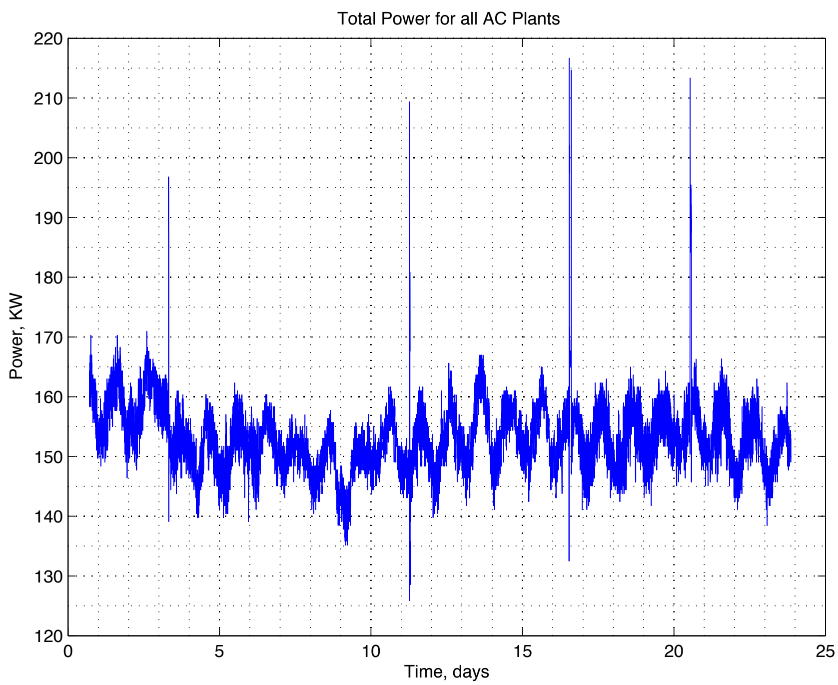

Figure 31: Time Series of AC Plant Cumulative Load

From this time series, it is evident that the loading profile has a couple of notable features. The first is that the transient periods of switching AC plants yield power spikes periodically, as discussed previously. The second is that the AC plant loading is heavily diurnal; over each one-day period there exists a minimum that occurs in the early portion of the morning and a maximum that occurs in the afternoon.

The diurnal behavior presents an additional difficulty for a stochastic model. Since the AC plants appear to have a time-variant loading profile assigning random variables to each AC

plant would not reflect the true behavior. Rather, it would be expected that when load is high for one running AC plant it would be high for all AC plants. These effects would need to be addressed for the stochastic model to provide realistic simulation results.

3.2.3 Trials Data Stochastic Analysis

A new feature included in the most recent revision of DDS 310-1 is guidance in adjusting an EPLA based on the sea trials data of a newly constructed ship. In addition to providing an update, the component monitoring performed during these trials provides a chance to develop or improve stochastic models. The trials data for the USS SPRUANCE used in this thesis provides an example of how simply this could be performed. Revisiting the operation of the lube oil purifier heater, shown previously as a time series in Figure 27, the heater exhibited a varying level of power consumption over time. This behavior, controlled thermostatically, did not seem to follow any regular pattern. By analyzing the data as a distribution, however, the data can be examined as a PDF, shown in Figure 32.

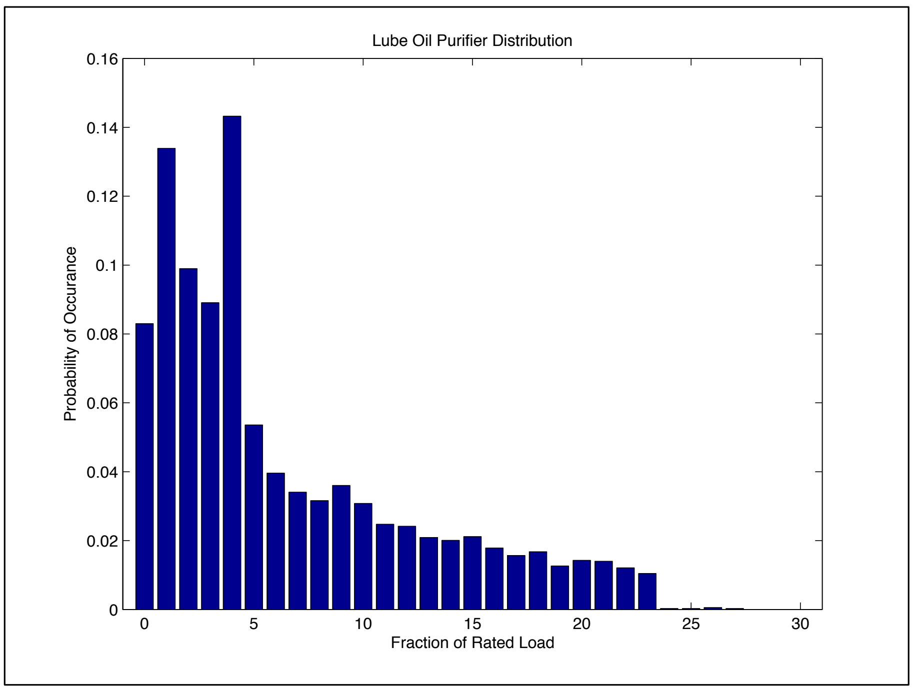

Figure 32: PDF of Lube Oil Purifier Heater Operation

This demonstrates how readily available data from the construction sea trials for a ship could quickly be used to develop a database of stochastic profiles. Proper planning would be required prior to conducting this analysis to ensure that the data series taken for different components captures the fully variability seen in the load to ensure proper PDF development.

3.2.4 Stochastic Analysis Summary

Stochastic analysis provides a more robust method of determining ship's electrical demand load than load factor analysis when computing an EPLA. Unlike load factor analysis, this method is capable of producing output statistics and developing relevant estimates for parameters such as peak loading. There are potential pitfalls that could occur using this method, however, as shown in the AC compressor example. The following section of this thesis outlines a method of performing behavioral analysis, which is in essence an extension of the stochastic models shown here. The framework of this analysis would utilize system behaviors instead of the random Monte Carlo method to perform the simulation. The implementation for a behavioral model may be different than the stochastic simulation, but the capability to define the relevant statistical information for a system or component remains a vital consideration.

4 Behavioral Modeling of DDG-51 Class

The behavioral modeling approach outlined in this section of the thesis incorporates many of the elements derived in previous sections to define a new method of performing an EPLA. Fundamentally, this model allows a user to define system responses to global inputs and uses statistical models to predict component electrical demand within the system. This method of performing an EPLA could provide an adaptable model capable of increasing in fidelity as the ship design process progresses.

4.1 Introduction and Motivation

Design efforts for US Navy ships typically begin many years before the first ship will be commissioned. The original DDG-51 design efforts, for example, commenced in the late 1970's, the first ship was contracted in fiscal year 1985, and the DDG-51 was commissioned in 1991 [29]. This long timeline creates a problem for the ship designer; equipment central to a ship's mission may not even exist when aspects of the ship, such as electrical generation capacity, must be determined.

An alternative to the detailed modeling and simulation method of performing an EPLA would be to create models that use simplet methods to provide a realistic time series approximation of data. Unlike the modeling and simulation approach shown previously, this method of modeling would not be based on developing component models from the underlying equations governing their electrical properties. Instead, this method would use statistical quantification of expected loading properties, not unlike those shown in the stochastic simulation example to develop realistic load profiles over time.

To demonstrate the behavioral approach a model of the 1SA switchboard was developed. This switchboard provides the primary source of power for the forward lower portion of a DDG-51, and its location in the ZEDS can be seen in Figure 4. The 1SA model was jointly constructed with Katherine Gerhard and Uzoma Orji at MIT.

The goal of this more limited model was to demonstrate the functionality envisioned in a ship-wide behavioral model, examine limitations presented by the method, and compare

the accuracy of the model to actual fleet data. Systems were modeled across the entire ship, as interdependencies cannot be truncated to only those components on this switchboard. The resulting load outputs were modeled, however, only for loads connected to the 1SA switchboard. Due to the total number of loads onboard a ship, using the 1SA switchboard limited the data input required to a manageable size while retaining the demonstration capability.

The 1SA switchboard is the largest switchboard on the ship, with more than 20% of the ships connected load and more than 25% of the expected operating load in a cruising condition [30]. The 1SA switchboard also represents most of the forward main engine room's source of power, and the engineering plant contains many of the loads most readily modeled. For these reasons, the 1SA switchboard was selected as the focal point for model development.

For real-world comparison, the selection of the 1SA switchboard also has the benefit of being monitored in MCMAS. Referring to the electric plant diagram in Figure 4, it is seen that all power consumed by the 1SA switchboard must pass through either the 1S-1SA or 1SA-2SA breakers. Since readings for the current, voltage, and frequency can be determined from the MCMAS program at these locations the power consumption can be estimated using the average power factor seen at the generators (which have the above indications, as well as power). Graphically, a representation of approximately 3 weeks of load for these breakers and the 1SA switchboard can be seen in Figure 33.

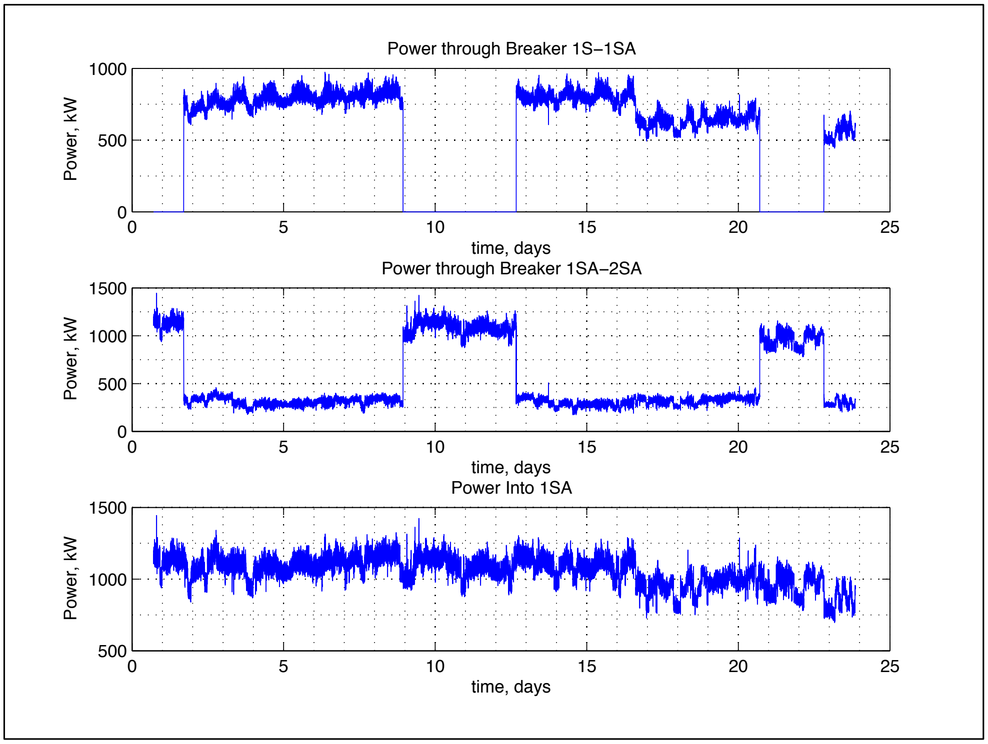

Figure 33: 1SA Loading Profile

The power seen through the 1S-1SA breaker drops to zero during periods when GTG #1 is secured, and based on the changes in power levels when this occurs it can be seen that the power in the switchboard is the summation of the two inputs sensed at the breaker.

4.2 Ship Operational Concept

The purpose of a naval vessel is to provide a platform capable of meeting required mission objectives. The ship's mission at a given time will dictate a certain set of operational equipment to be in use, which will directly impact the electrical generation capacity for the vessel. The creation of a fully functional model, therefore, must be responsive to the input signal variability caused by changing operating conditions.

The operating conditions selected for the behavioral model are based on those defined in the DDS 310-1; anchor, shore, cruising, and functional. Based on available data, the cruising operating condition will be evaluated in this model.

It is important to note that engine configuration and ships speed are coupled; there are certain speeds that cannot be achieved without a specific engine configuration. The overall profile of engine configuration and operating speeds should ultimately reflect the actual operating profile of the ships, as was shown in section 2.1, if model is to be reflect current fleet performance. For the purpose of the model development, actual data sets from DDG-51 class ships were used for the engine configuration and speed profile to ensure accurate inputs.

For design studies involving future ships the engine configuration, ship speed, and mission should all be reflected through the concept of operations (CONOPS) for the vessel. The CONOPS reflect the expected operational utility the vessel will have within a battle force, and the expected utility it should provide. These documents are developed from the point of view of the individual or organization who will be operating and utilizing the asset.

Bart Sievenpiper 4/24/13 8:29 AM
Comment [3]: Need better def.

An example of how the concept of operations is used to develop can be found in examining the concept of operations released in September 2012 by the US Coast Guard for the Offshore Patrol Cutter (OPC). This document was released as part of the request for proposals (RFP) for the design of the next-generation mid-sized cutter in the Coast Guard fleet. This document handles everything from the required operating areas the ship is expected to utilize to the missions it will undertake. The expected mission profile for OPC is shown in Figure 34.

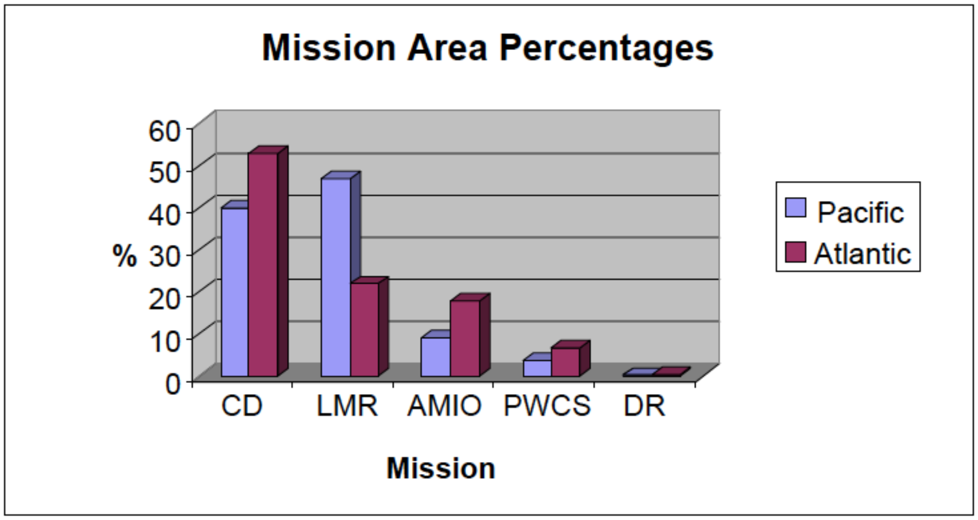

Figure 34: CONOPS Mission Percentages for OPC [31]

In this graphic the first mission depicted is counter-drug (CD), which is the pursuit of narcotics traffickers in maritime environments. Living marine resource (LMR) is the mission of protecting coastal waters from foreign encroachment and enforcing fisheries laws. Migrant interdiction (AMIO) is prevention of unauthorized entry of foreign personnel into the country via boat. The ports/ waterways/ coastal security (PWCS) and defense readiness (DR) missions include the potential use of force to protect American assets domestically and abroad. Each of the missions would require a different subset of shipboard equipment to perform the ship's functional duties. When output results are properly tied to input parameters, the ship designer should be able to look at Figure 34 and answer a question like: "How will predicted electrical loading vary based on the different mix of Atlantic and Pacific mission assignments?"

Beyond missions, the speed-time distribution is documented in the CONOPS for the OPC. As discussed previously, these speed-profiles will directly effect electrical consumption based on dependencies established in the engineering plant. The expected OPC profile is shown in Figure 35.

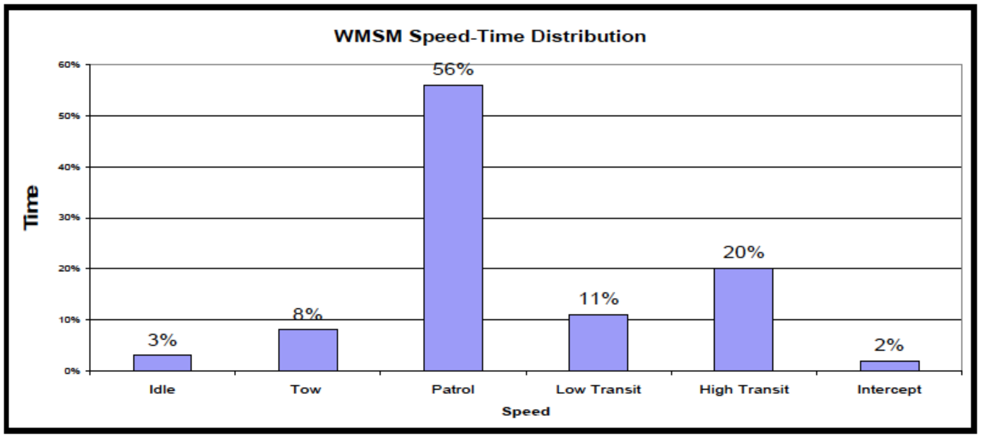

Figure 35: CONOPS Speed-Time Profile for OPC [31]

If one wanted to evaluate the trade space for propulsion to select between IPS and conventional diesel or gas turbine drive, having an adaptable design tool to model fuel consumption would be needed. Current EPLA methods could have difficulty accounting for electrical propulsion loading effects.

These graphs from the OPC RFP demonstrate a typical series of early-stage definitions. They also highlight the types of questions that current EPLA methods do not address. Model development must ensure that future processes can interface cleanly with external input parameters to provide useful results.

4.3 Model Framework

To develop a model for electric plant design, the first challenge was to develop an interface that would be robust enough to demonstrate the potential utility of the proposed model. The system was first required to accept representations of the external inputs, which represent the operational profile for the ship. This includes both the direct drivers onboard the ship, such as speed and engine profile, and the external inputs such as ambient temperature. The system characteristics must then be defined for each component in a manner that reflects their dependency on these external inputs. The component responses, and associated electrical signature, must then be integrated into the model to ultimately create a realistic simulation.

In this process system behaviors are decoupled from electrical responses. An example of how the power simulation is created is shown in Figure 36, which depicts a notional system component. In this example the top image is the square wave operating state for a component that is modeled based on system behaviors. This is the "ones and zeros" representation for the component, while the center image depicts the transient electrical response seen when energizing the component. By superimposing the electrical behavior of the on the square wave profile, an estimated power trace for the component is developed.

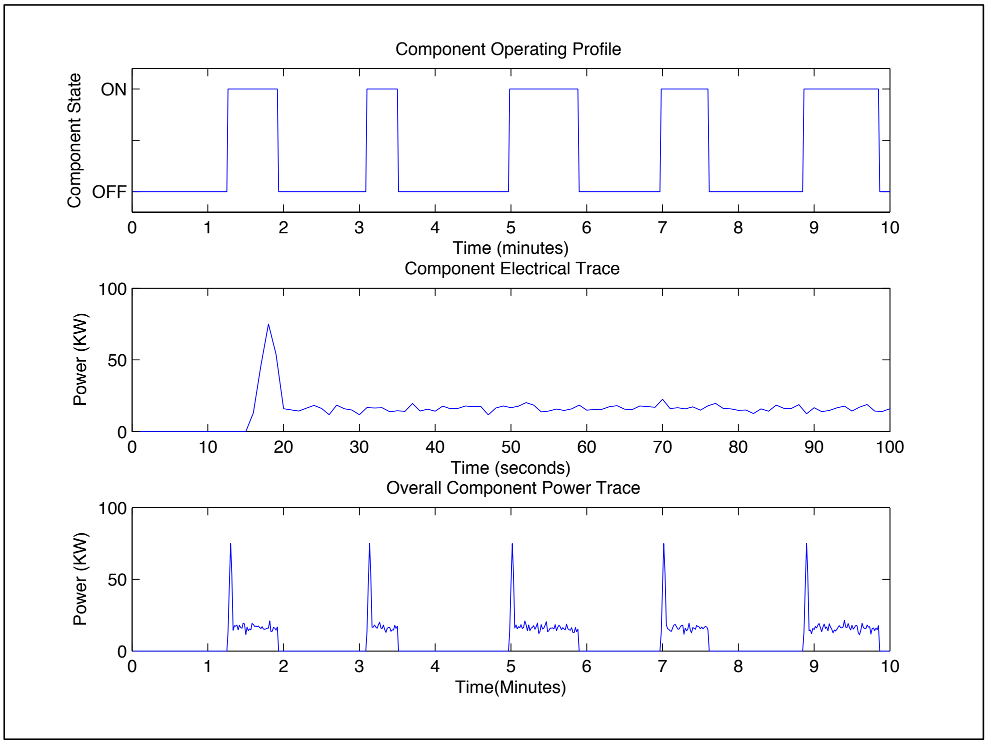

Figure 36: Method of Creating Power Trace

This creates model that is adaptable over time, as information is gained on how either the system behaves or how the component responds improvements can be made independently of other variables.

Overall, the process of developing a model that takes global inputs and develops an electrical model is shown graphically in Figure 37. Each of the steps involved in developing this model will be shown in the following sections.

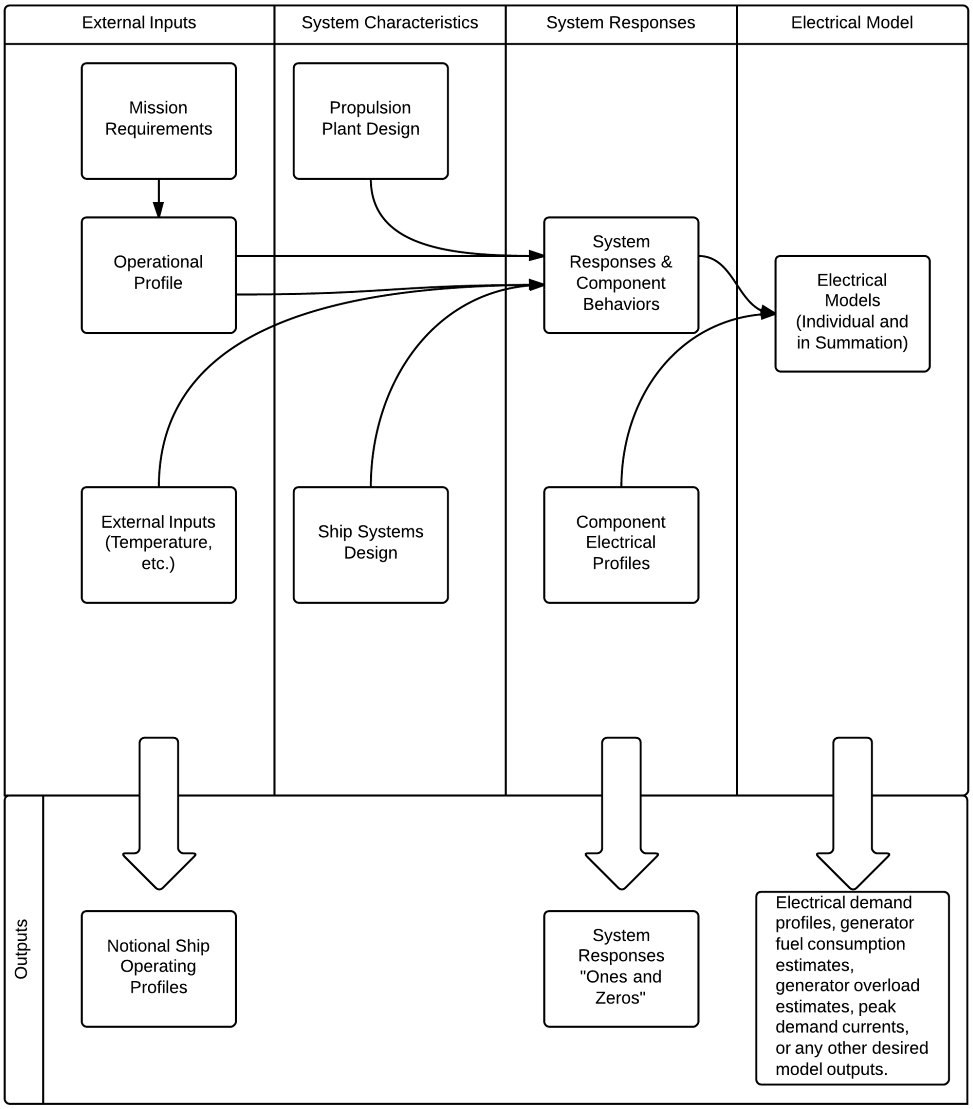

Figure 37: DDG-51 Model Design Process

It is important to note that the model developed ignores an introductory screen in which the basic ship variables required for a model are defined. This screen would allow the ship designer to define the propulsion plant architecture, electrical architecture, and other ship parameters of interest. By using a current ship class, this information is well defined. The major parameters representing this ship information are shown in Table 8.

Table 8: Ship-Type Definitions

Ship-Type Definitions	
Variable	DDG-51
Ship Classification	Surface Combatant
Propulsion Type	Gas Turbine
# of Shafts	2
Electrical Generation Type	Gast Turbine
Electrical Distribution Type	Zonal
Combat Systems	Aegis
Auxiliaries	Defined

4.4 Global Inputs

Global inputs, as defined in this thesis, are those parameters external to the ship that drive the behaviors of systems within the ship. The global parameters will not generally change for different ship types, as they impact the ship regardless of the definition of systems onboard. The global inputs evaluated in this model were propulsion plant operating mode, ship speed, time, season, and ambient temperature. For demonstration purposes in this model, the data used was actual operating data for a deployed DDG Flight IIA ship. As a result, no model was created to generate the global values, since the inputs were well defined.

The model developed to demonstrate this concept required a series of inputs prior to running a simulation. The initial screen operated by the user is shown in Figure 38. In this model, three selection buttons are provided on the main screen; one for modifying the environment-type inputs, one for the ship-type inputs, and one for model parameters. Once all selections have been made, provisions are made for the running the simulation and acquiring output results.

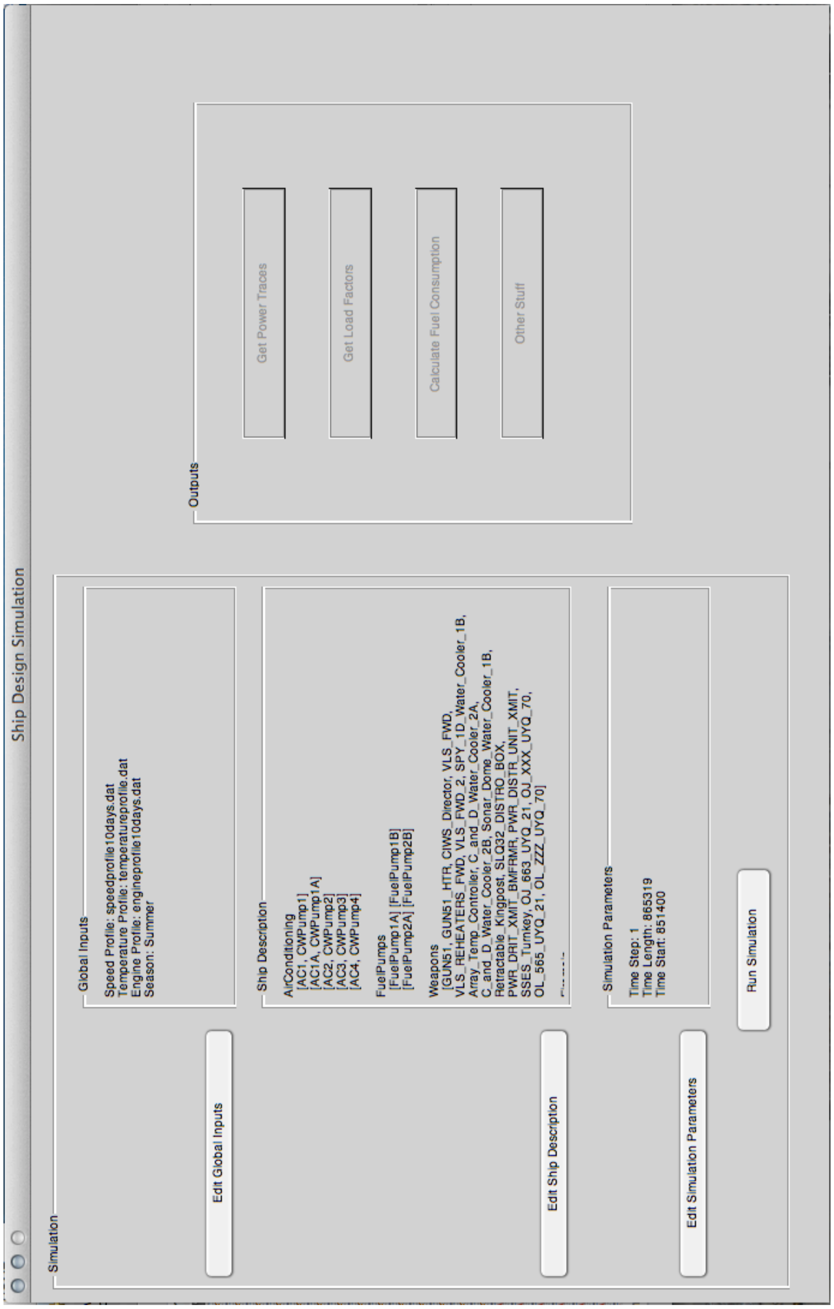

Figure 38: Graphical Main Interface Screen

Selecting the global inputs button brings up a menu allowing the input of several different parameters for the simulation. In the framework of this program this is set up to accept these parameters as input files, with the menu shown in Figure 39.

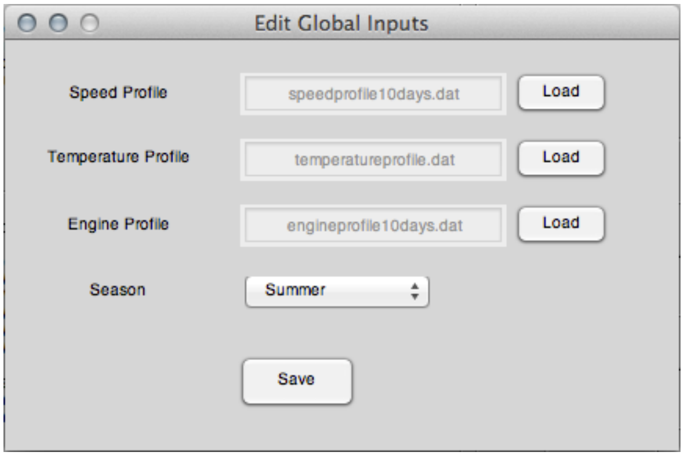

Figure 39: Global Input Menu

In this model the speed, temperature, and engine profiles are entered in as data files. The selection of the operating season aligns the model with the current data incorporated in the EPLA, which separates winter and summer operations. A more fully integrated model would utilize the stochastic input requirements, such as the speed and mission profiles shown for the OPC previously.

4.5 Modeling Ship System & Subsystem Behaviors

Entering the ship description requires definition of each component within the ship. With thousands of individual loads onboard the ship, ensuring this is performed in a logical manner is crucial. For the purposes of modeling, this is best accomplished by organizing the ship into systems, subsystems, and loads. The purpose of separating the ship in this manner is that the systems and subsystems govern behavior of components, while the electrical response of the load. A simple example of this hierarchy is the electrical generation system. In this case the system level governs that two of three GTG sets will be operating, and the operating generator will switch with a known periodicity. The subsystem for the GTG set governs how the loads within the subsystem behave,

particularly that when the subsystem is operating the cooling water pump and cooling fan will be energize and the enclosure heater will be secured. The component definition provides the definition for how the component will respond within the model when the subsystem indicates that it is running. In some situations, where a load operates independent of other loads no subsystem would exist, and the system design would directly control the operation of the component.

The interrelations between the system, subsystem, and load can also be seen graphically, as demonstrated in Figure 40. In this case the air conditioning system is broken into 5 subsystems, with each subsystem operating two loads.

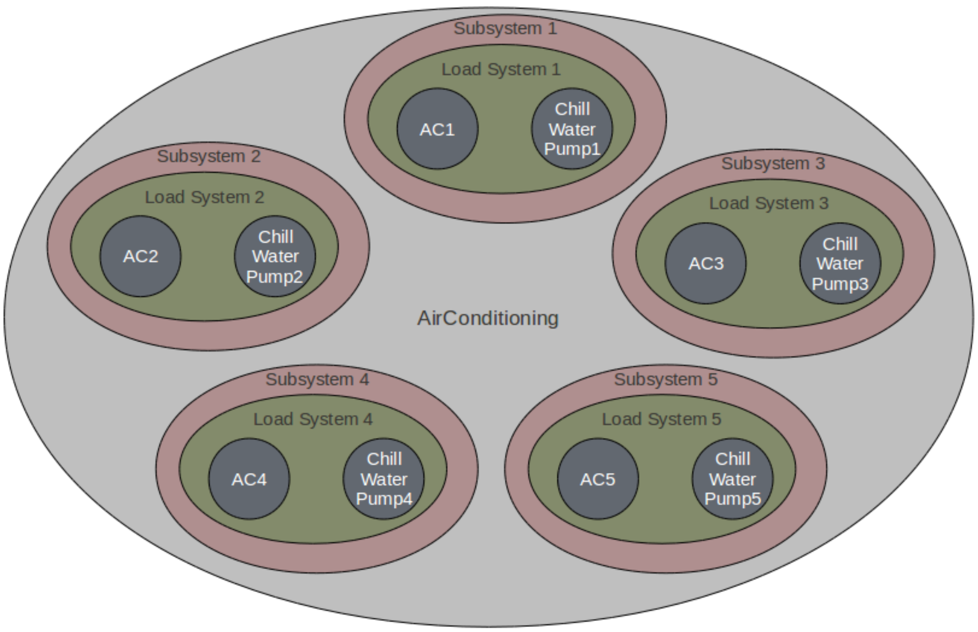

Figure 40: AC System Relationships

The first step in the load definition therefore is the screen that provides a means of entering ship systems. This entry page is demonstrated in Figure 41, and shows a sample list of systems. The system loads, grouped by specific subsystem, are shown in a secondary window when the ship system is selected.

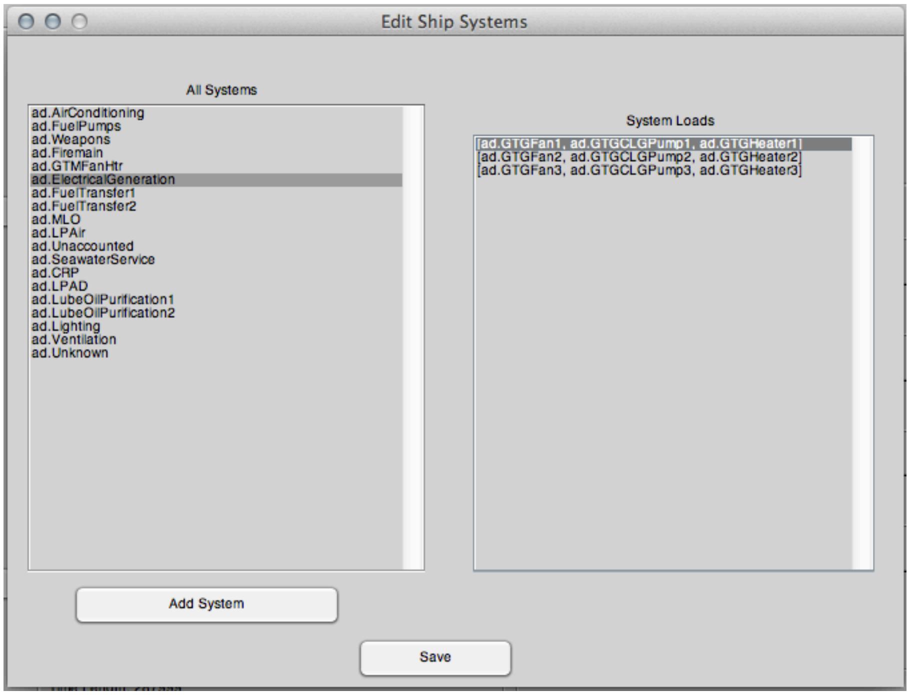

Figure 41: Model Ship System Entry

When the “add system” button is selected, it brings up the option to build a new system into the program, as shown in Figure 42.

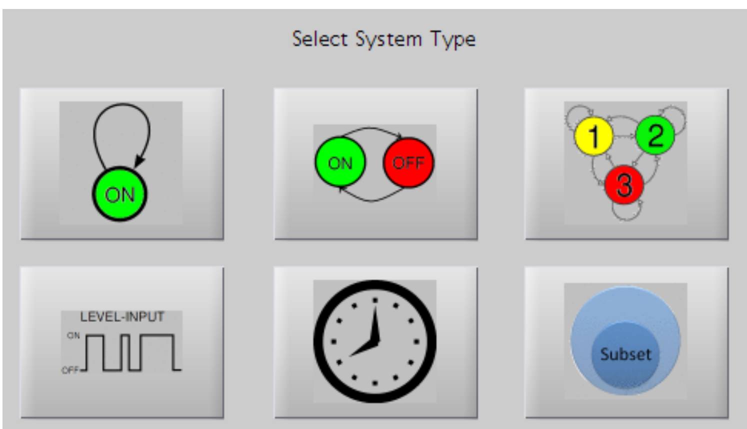

Figure 42: System Entry Selection Menu

When a new system is entered, it will require definition first based on the type of system that it represents. In developing the model for the DDG-51 six types of models were identified, each of which will be discussed in the following sections. These model types are general enough that they describe both the system and subsystem behaviors, as the selection of an operating subsystem may behave differently than the method in which the subsystem operates individual loads.

The framework was implemented in the Matlab programming environment using an object-oriented programming data structure. The MATLAB code for the system object is shown in ShipSystem.m, which is located in Appendix A.1. The ShipSystem object is responsible for generating an operational profile for all of its subsystems by calculating the intervals of times in which each subsystem is on or activated. The six models identified to calculate these time intervals are discussed below.

4.5.1 Single-State

The single-state condition is the simplest to define, and simply refers to a system or subsystem that maintains a single configuration in a given ship state. As an example, this could apply to ventilation fans, radars, or communication equipment that is running continuously during ship operation. It could also apply to equipment, such as a main reduction gear turning gear, that is secured during all underway operations. This equipment would use the input state of the ship to dictate the configuration of the system.

4.5.2 Cycle Type

A system with cycle-type characteristics behaves periodically, but independent of the time of day. These systems often have on/off cycling behavior that is somewhat predictable. An example of this is the cycling behavior is the lube oil purifier onboard a ship. The purifier runs (as was shown in Figure 17) with a regular periodicity that can be readily defined. Understanding the distribution of the time the purifier is running and the length of time it is stopped, this piece of equipment can be readily categorized. The stochastic models that govern the length of time the system remains in the on or off state require user inputs. These behaviors can be determined using the methods developed for stochastic modeling, defined in Section 3.2.

4.5.3 Finite State Machine

The final way designed to model system or subsystem behavior is through the use of finite state machines (FSM). A finite state machine allows the selection of operational conditions based on probability of transitioning between various states. A simple example of a FSM is shown in Figure 43. This example represents a system with three states: off, low, or high. At a given time step the system can remain in the same state or undergo state transition, with each possibility being defined with a given probability. As shown below, this system is unable to transition directly between the off and high state, but rather must pass through the intermediate low state. For any system that uses an FSM definition, it would be imperative to properly define all possible transitions and the probability associated with each.

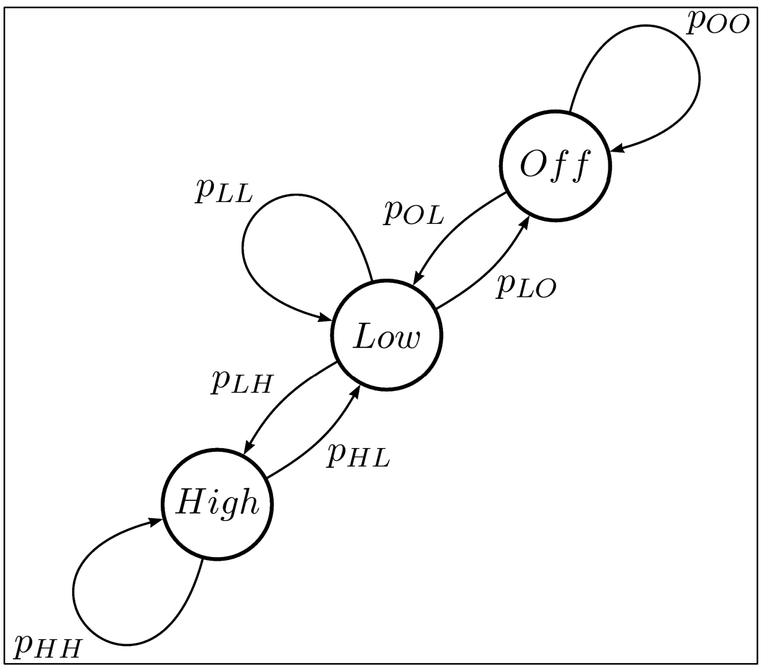

Figure 43: Finite State Machine Example

In this example the machine can be off or in one of two states, high or low speed. To achieve high speed, however, the component must be transitioned through the low speed

setting. At each time step in a model the component have the potential to remain at the same state it was in the previous time step, or to make a transition to a different state.

The user defines the stochastic models of each state and the overall transition probability matrix. For the FSM in Figure 44, the corresponding probability matrix P is shown in Equation 3. Through the use of the tranisition probability matrix, the model will randomly select the next state at each step.

$$P = \begin{bmatrix} p\{OO\} & p\{OL\} & 0 \\ p\{LO\} & p\{LL\} & p\{LH\} \\ 0 & p\{HL\} & p\{HH\} \end{bmatrix}$$

Equation 3: Probability Transition Matrix

4.5.4 Level-Type

The level-type system is directly dependent on the state of a global input. In this case the system may be on whenever a specified condition is met, and secured during all other conditions. An example of this is the fuel service system, in which a pump for a specific plant will be energized whenever one of the two GTM in the operating plant. In this case, the subsystem is dependent on the "level" of a global input, shown graphically in Figure 44.

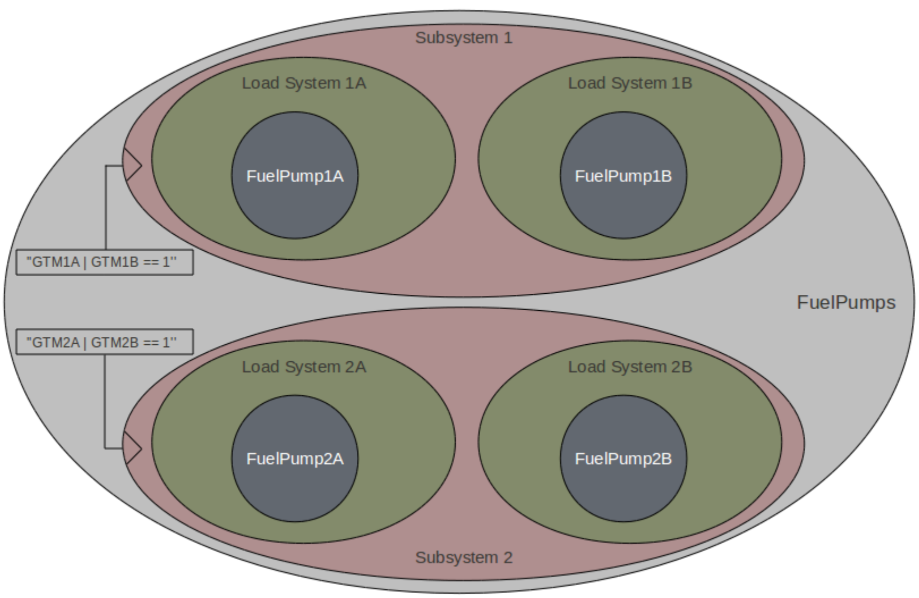

Figure 44: Fuel Service System Model

4.5.5 Time Dependent

The time-dependent system depends on the time variable of the model to drive the cycling performance of the system. Some systems operate in a predictable manner over the course of a day, or periodically over the course of several days. Food service equipment in the galley is operated during meal hours, and sparsely during other times of day. Systems with these time-dependencies are modeled to energize with varying frequency based on the time of day. The user interface would provide a flexible design environment where the user can define the temporal interdependencies required to properly model the system.

4.5.6 Random Subset

The random subset method is necessary as Navy ships are constructed with a large amount of redundancy. This redundancy drives the addition of multiple subsystems that can all perform the same function equally. The random subset method will select the required number of running subsystems, and at a defined periodicity switch the running subsystems by selecting a new random subset. The AC plants, as shown in Figure 40 previously, were modeled in this manner at the system level.

Ensuring these systems are design correctly is vital to ensuring a realistic output is achieved. An example of this is the ship's auxiliary systems. While not the first thing that might come to mind when defining a Navy ship, auxiliary systems are the largest group of electrical consumers onboard a ship. Of the electrical energy used on a DDG-51 class ship, approximately 45% alone comes from the heating, ventilation, and air conditioning system [32]. The auxiliary systems incorporate everything from firemain to sewage processing, and encompass many independent systems that each have a unique design. In the case of the DDG-51 the systems are well defined, but for a new ship design effort each system would have to be designed in the model.

4.6 Load Electrical Modeling

Once the behaviors of components has been developed through the system and subsystem definition, a means of modeling the electrical response must be implemented. Within each

subsystem all subordinate components must be defined by the user. The MATLAB code for the component object is shown in ShipComponent.m, which is located in Appendix ##. The user first defines each component as a master or slave. A slave component has the operational profile of its subsystem. When the subsystem is active, the component is always on. A master component is on when its subsystem is active, but has its own operational profile described by a stochastic model that is defined by the user. The various types of stochastic models are described in Section 3.2.

Depending on the operating procedure of the system, the user may define offset parameters. These values can be used to prevent components from turning on at the identical time, which could cause erroneously high transient behavior. Onboard a ship, operators would perform steps sequentially instead of the same time. For example, in the Electrical Generation system, when the corresponding GTG turns on, the module cooling fan turns on first. Within a couple of minutes, the cooling pump turns on and several minutes later, the heater turns off. These steps all occur prior to starting the GTG. The turn-on offset parameter for each component can be set accordingly to follow this operating procedure. It should be noted that the heater turns off when the subsystem is active and turns on when the subsystem is inactive. The user can define a flag parameter to invert the operational profile of the component in comparison to its subsystem.

The ShipComponent object (contained in Appendix A.2) is responsible for generating the electrical response based on the operational of its parent subsystem. For this model, four separate methods of implementing a component response were used: constant, finger-print, finite state machine and time-dependent as shown in Figure 45.

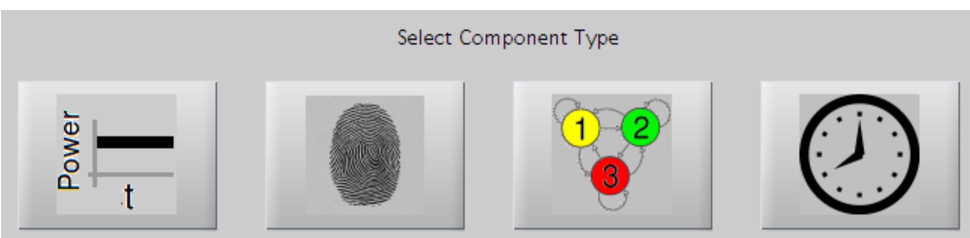

Figure 45: Load Modeling Selection Screen

Each of these methods provides a unique means of developing a power trace, and is expanded upon in the following sections.

4.6.1 Constant

In a load factor analysis derived EPLA the electrical utilization is accounted for using the rated (or nameplate) load of the component and multiplying it by the load factor for a given condition or season, as shown in Equation 4. This method was described in significant detail in Section 1.4.1.

$$P_{avg} = LF \cdot P_{rated}$$

Equation 4: Load Factor Estimation of Power Consumption

Many loads that do not have defined electrical behavior, and using the load factor assumptions may remain the best approach. Once the electrical (or potentially system) behaviors are determined, the load could be modeled using a higher-fidelity method.

Incorporating load factors from previous EPLA for every load onboard the ship the user can compare the results of a simulation to those that would be obtained from the load factor analysis. In cases where the model deviates significantly in its long-term statistics from the load factor results, this can provide a means of determining the sources of error.

4.6.2 Fingerprint

The fingerprint method assumes that the load follows a unique and regular behavior that consists of three phases: a transient turn-on phase, a steady-state phase, and a transient turn-off phase. Each phase is individually defined within the model. This method is most useful for components such as motors that exhibit regular behavior, and is most applicable when the operating profile for a component is readily available. With the data available from the baseline of the USS SPRUANCE, many of the components on the ships have available profiles. An example of how this data looks graphically is shown in Figure 46.

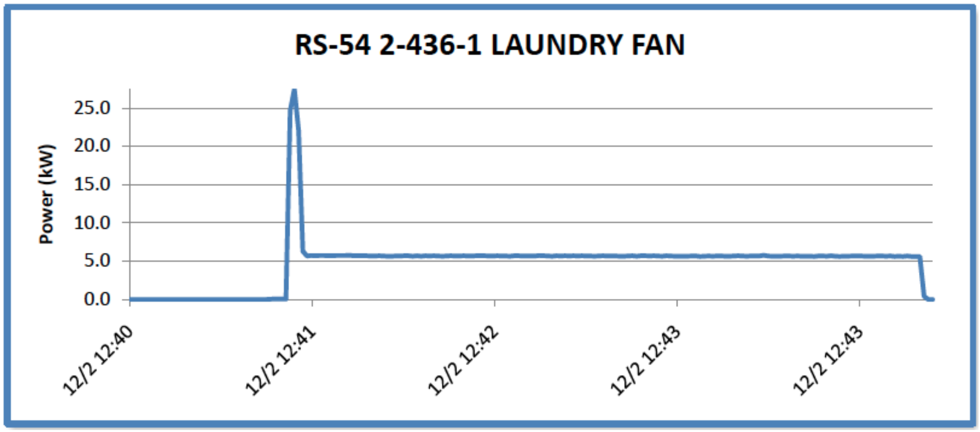

Figure 46: Example Electrical Operating Data [32]

From this figure it is straightforward to determine the three ranges of the operating cycle. The start-up transient is a short, well-defined inrush spike lasting only a few seconds. The steady-state region is clearly a well-behaved one for this fan, with little variation in the operating load for the component. The turn-off phase when the component is secured is again a short (order of seconds) transient that can be easily defined.

Each region of the load's operation is defined and stored within the model, and can be applied to the based on the condition of the load. When a load starts, the transient behavior will run through the allotted number of time steps. Following this, the steady-state behavior takes over until the component is secured. Steady state behavior can be modeled either as a looped series that recreates the actual variation seen in the equipment, or modeled as a stochastic distribution.

A large fraction of the HVAC, auxiliary, and engineering loads onboard ships utilize induction motors to perform their function. The profile for these motors is very similar, regardless of the function they serve. The profile looks like that shown in Figure 46, an initial starting current of approximately 5-7 times the running current followed by steady operation. If the load demand changes for the motor, some variation will be seen in the steady region. Incorporating product models for common equipment, like induction

motors or resistive heaters, would simplify the process of developing a model for a future ship design.

Bart Sievenpiper 4/24/13 10:12 AM
Comment [4]: Does this warrant another section?

4.6.3 FSM

The finite state model is similar to that described in the fingerprint model, but requires a more complex series of inputs. For subsystems with different operating levels or conditions the FSM model was utilized, as described above. When the FSM model is utilized, however, it is necessary to define an electrical profile for each operating state of the equipment.

For most types of equipment, the electrical profile will be similar but at a different power level, though this does not have to be the case. In each operating state it would be possible to define the electrical consumption using any of the other methods described here.

The NILM data from collected by Bennett at LBES demonstrates the effects for fuel oil service pumps, shown in Figure 47. These pumps are positive displacement two speed pumps, and it is evident that when the pump shifts from low to high speed, doubling pump flow rate, the running current approximately doubles. This is the expected behavior for a positive displacement pump, but demonstrates the need for unique fingerprint models in each state.

For components described by an FSM, the user must define the stochastic model for each state and a transition probability matrix like those discussed in Section 4.5.3.

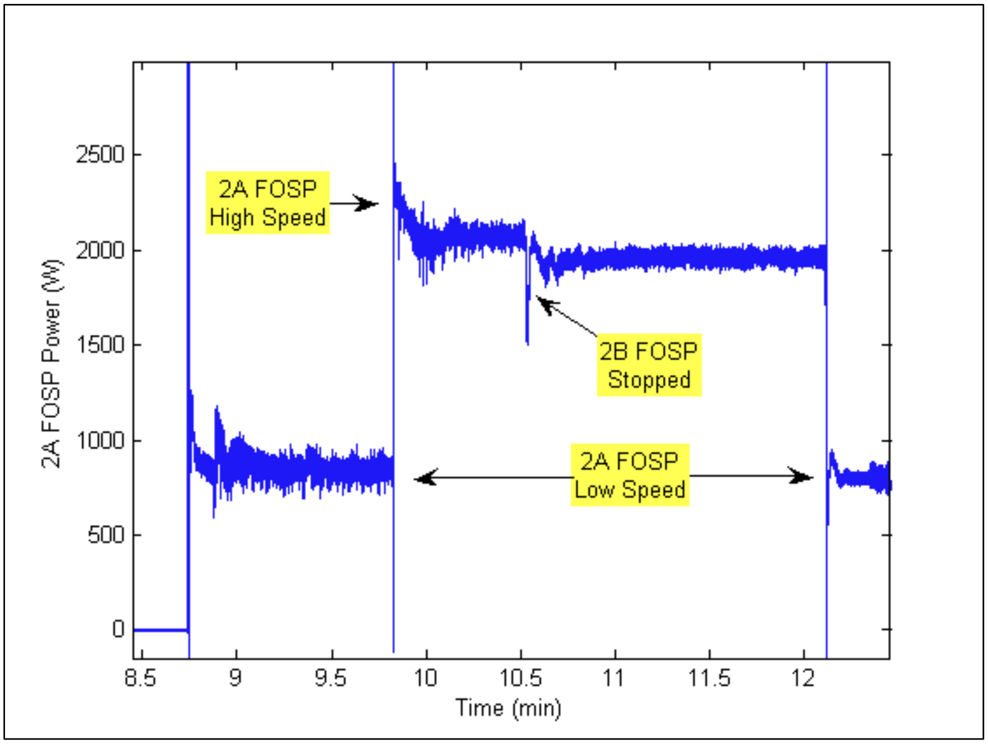

Figure 47: LBES Power Drawn by 2A FOSP During a FOSP Pump Shift Evolution [20]

4.6.4 User Defined

The final profile is one required for input dependent systems. In this case, one of the input variables for the model drives the electrical profile of the load. When using this method the user defines the interrelations between global inputs to the model and component electrical response. This allows the creation of dynamic component profiles that simpler methods cannot replicate.

One example of this is the AC motor compressor, which previously was shown to operate with strong diurnal behavior. When developing a PDF for the AC compressor in the stochastic simulation section (Figure 29), it was noted that the overall electrical consumption profile could be obtained but would lack the characteristics of the actual loading profile. To examine these temporal effects, the data was sorted such that individual

profiles were obtained for each 2-hour block of time over the course of a day: 0000-0200, 0200-0400, etc. By taking the mean and standard deviation of all individual AC plants during each time segment using several weeks of data Figure 48 was created.

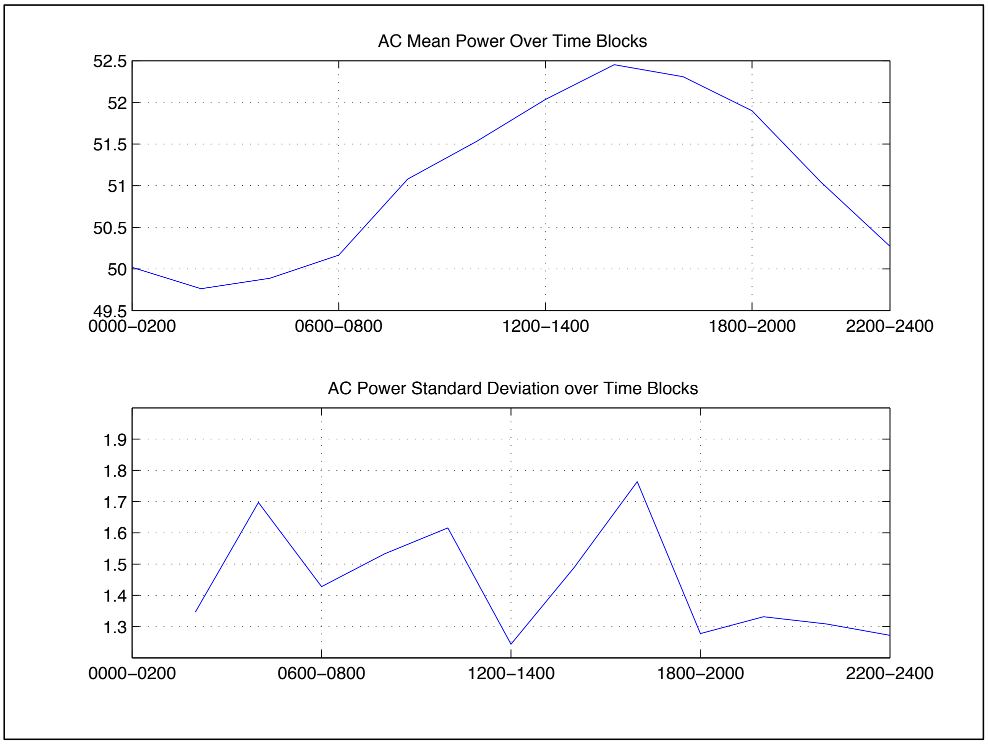

Figure 48: AC Compressor Power Mean and Standard Deviation

Additionally, the AC system loading would be expected to change with the day-to-day variations in ambient temperatures. During the time period of analysis the vessel is operating in a single location performing a continuing presence in anti-piracy operations. This singular mission profile allowed the investigation the effects of ambient temperature. By plotting the daily high, low, and mean temperatures seen in cities near the ship's operating location in the Gulf of Aden the weak effects of the temperature can be seen in Figure 49.

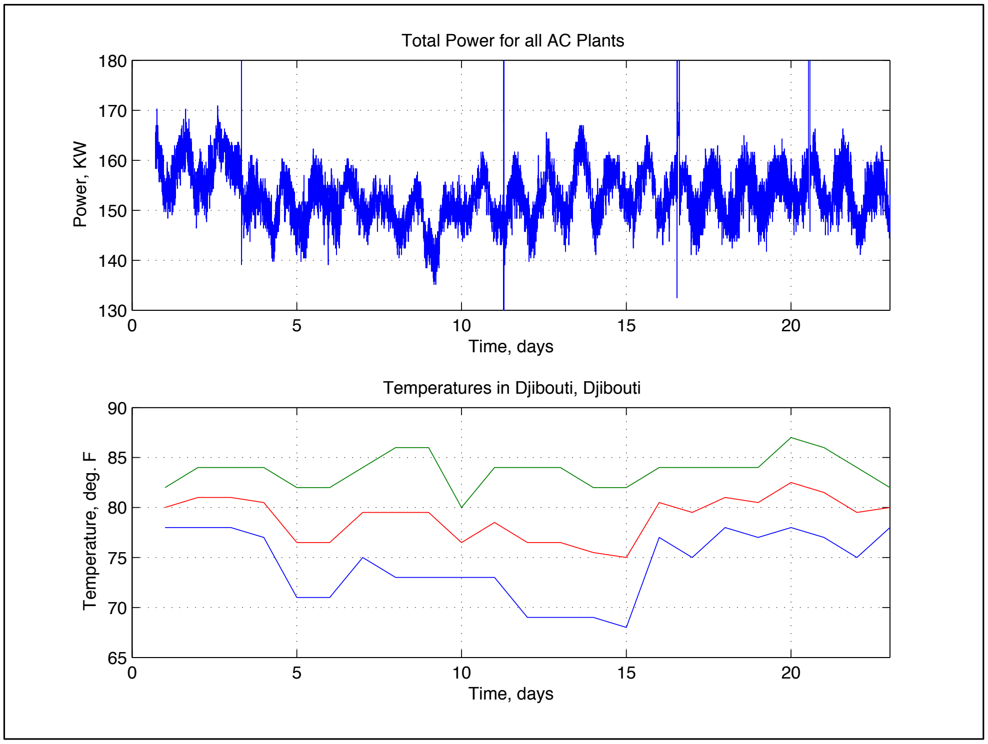

Figure 49: Total Ship Compressor Loading and Temperature Variation

Examining this data, it was noted that there was an additional correlation between the mean value for temperature and the total load placed on the AC compressors. By allowing the mean shown in Figure 48 to drift slightly with the variation in temperature a more accurate behavioral model could be developed.

For this model, the user can input parameters (A,B,C,D,E) to model the effect of the global input. For the AC compressor, the relationship between the temperature, **T**, and the temperature dependent mean, μ(**T**), is modeled as shown in Equation 5, where the remaining variables are fit to the data set.

$$\mu(\mathbf{T}) = \frac{(A(\boldsymbol{T} - B)^C + D)}{E}$$

Equation 5: Base Model Expression for AC Compressor

In a preliminary design of a ship many of the interrelations needed to develop these loads may not be available, and fidelity would be expected to improve as design progresses. This electrical definition process does, however, allow a means of incorporating future capabilities, such as radar systems, in a unique way.

4.7 Stochastic Models

Stochastic models are used to describe processes with distribution functions that are known or can be estimated. These methods are used to describe the length of time a finite state machine remains in each state and the length of time for each interval a master component is in operation. In this framework, the five types of stochastic models utilized are constant, uniform, normal, exponential, and Poisson. Each of these will be described in the following sections.

4.7.1 Constant

The constant method represents the simplest possible distribution used in the stochastic models. In this model the user defines a value of **a,** which is related to the random variable X_c by

$$P(X_c = \boldsymbol{a}) = 1$$

Equation 6: Constant Distribution Representation

4.7.2 Uniform

In this distribution, a random variable X_U has a uniform distribution, $f\big(X_U(x)\big)$, if the PDF is constant within the interval **a** and **b**. In this case, the user defines the values of **a** and **b**, and the distribution is shown in Equation 7.

$$f\big(X_U(x)\big) = \frac{1}{(\boldsymbol{b} - \boldsymbol{a})} \; for \; \boldsymbol{a} \leq \boldsymbol{x} \leq \boldsymbol{b}$$

Equation 7: Uniform Distribution Representation

4.7.3 Normal

When a random variable X_N has a normal distribution the user must input the mean, μ, and standard deviation, σ. The system will then cycle with a frequency dictated by these values according to the distribution shown in Equation 8. Care must be used in implementing this distribution, to ensure the probability does not return negative time values.

$$f\left(X_N(x)\right) = \frac{1}{\sigma\sqrt{(2\pi)}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

Equation 8: Normal Distribution Representation

4.7.4 Exponential

When the exponential distribution is used to model a system, a random variable X_E has the distribution $f\left(X_E(x)\right)$ when the PDF is defined as shown in Equation 9.

$$f\left(X_E(x)\right) = \lambda e^{-\lambda \cdot x}, x \geq 0$$

Equation 9: Exponential Distribution Representation

When the exponential distribution is used, the user must input the variable λ, which is the rate parameter for the function. The choice of this value will determine how quickly the exponential function decays away, and will therefore influence the width of the distribution of cycle time.

4.7.5 Poisson

The Poisson distribution eliminates the concern with negative time values seen in the , and will return only positive time intervals. When this distribution is used a random variable X_P has a Poisson distribution, $P\left(X_P(k)\right)$, and creates a probability mass function (PMF) that is defined using Equation 10.

$$P\left(X_P(k)\right) = \frac{\lambda^k e^{-\lambda}}{k!}$$

Equation 10: Poisson Distribution Representation

In the case of the Poisson distribution, the variable λ represents the expected arrival time for event being modeled.

4.8 Behavioral Model Output and Results

Once all individual systems, subsystems, and loads applicable to the 1SA switchboard were defined the simulation was run. The most important aspect of the simulation for the calculation of an EPLA is the overall loading, since this will be the primary result used for the sizing of electrical distribution equipment.

The output profile for the 1SA switchboard is shown for a ten-day simulation is shown below in Figure 50, while the actual loading on the 1SA switchboard during this time period is shown in Figure 51. While simulation does not perfectly recreate the 1SA switchboard, it captures many features that exist within the system. Power transients associated with starting loads are captured, and much of the behavior over time is included. Since power transients have an impact on the sizing of generators, breakers, and cabling this behavioral model could allow designers to rely less on large margins and instead optimize the plant for expected load conditions. These results, using a relatively small subset of fleet data, demonstrate that the method can deliver high fidelity results and could enhance the ship design process. Overall, the simulation for the 1SA switchboard is bound within the same general region (800-1200 kW), indicating a good data fit for this simulation. Some components (primarily weapons and combat systems) did not have profiles that could be modeled from either MCMAS or the USS SPRUANCE data. For these components the previous EPLA values, calculated with load factors, was used, which led to producing a more level output than what is seen in the fleet.

The randomness of the system models dictates that no two simulations will be the same, and that the different power levels seen in Figure 50 will change for each run. Behaviors linked to inputs (such as GTM stops and starts) would be the same for every simulation. By running the simulation many times a long term statistical description of ship behavior could be created, similar to the process for a Monte Carlo method.

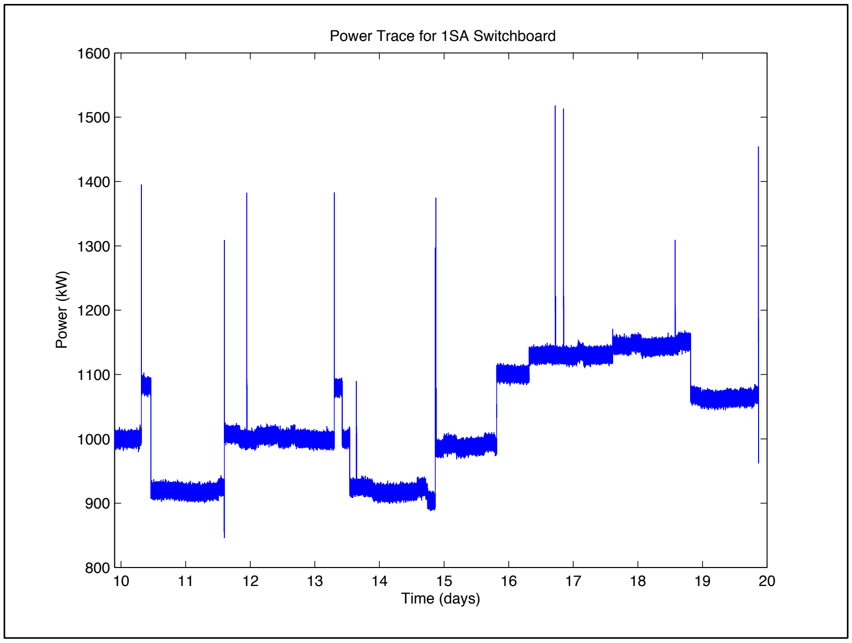

Figure 50: 1SA Simulation (10 days)

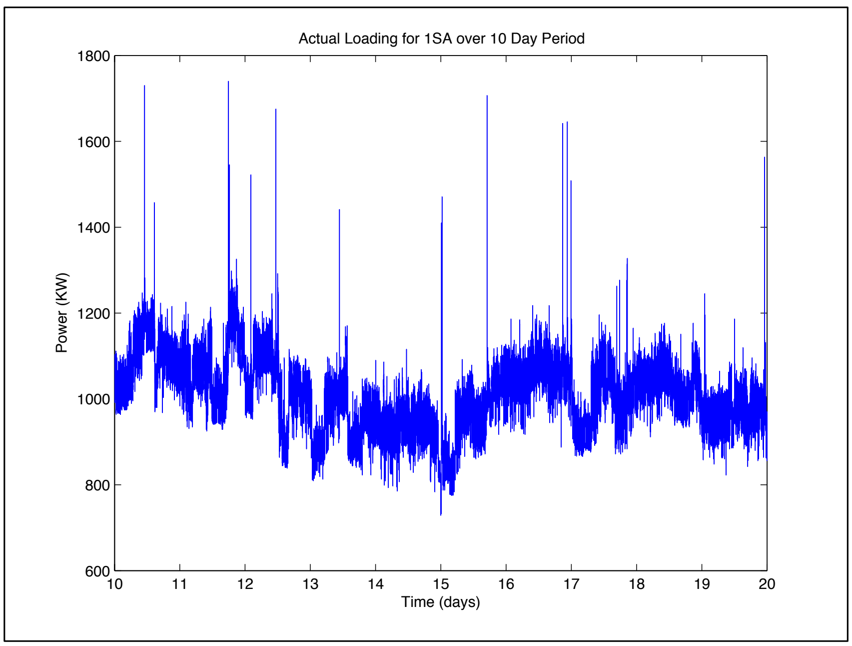

Figure 51: 1SA Actual Load (10 Days)

Within the program, results from this simulation are available to the user through an interface that allows examination of the results on both a component and overall basis. An example of the interface is shown in Figure 52.

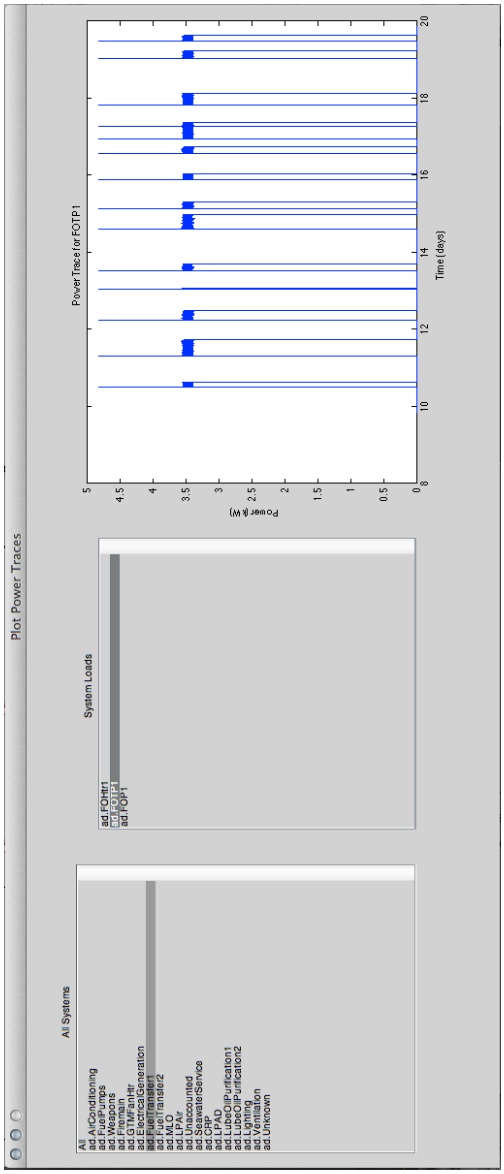

Figure 52: Model Output Window (Fuel Transfer Pump Selected)

This figure shows the selection of the fuel transfer pump #1, and the cyclic behavior it exhibited over the course of several days. This profile demonstrates how each component behavior can be seen individually as a time series within the program.

An additional benefit of using the program is that the individual results for a selected system could be analyzed if desired. This would provide the ability to use model results to inform selection of components, or could be used for the purposes of model validation.

Examining the AC system provides an example of how the system, subsystem, and electrical behaviors were implemented into this model. The random subset method of modeling the system ensures that three of five AC systems will be on at a given time, but also that the operating units will switch with a certain periodicity. Within the subsystem, two different components (the AC compressor and chill water pump) are operated simultaneously, one exhibiting periodic loading conditions and the other treated as an induction motor with stochastic definition. These behaviors are shown in Figure 53 and Figure 54.

The user-defined profile with diurnal variation is clearly visible in the model for the AC compressor over time. The unit is secured and restarted within the time period modeled based on defined system behaviors. No two simulations would have the same profile, since the configuration changes are controlled by a random distribution of time with a specified mean. The chill water pump exhibits the expected power peaking expected using the fingerprint method of load definition.

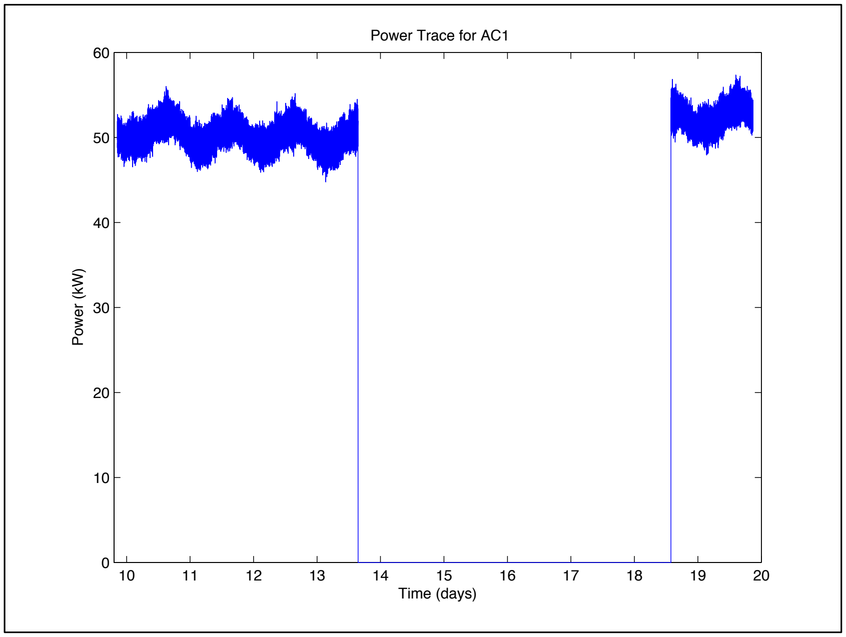

Figure 53: Model Results for AC Compressor #1

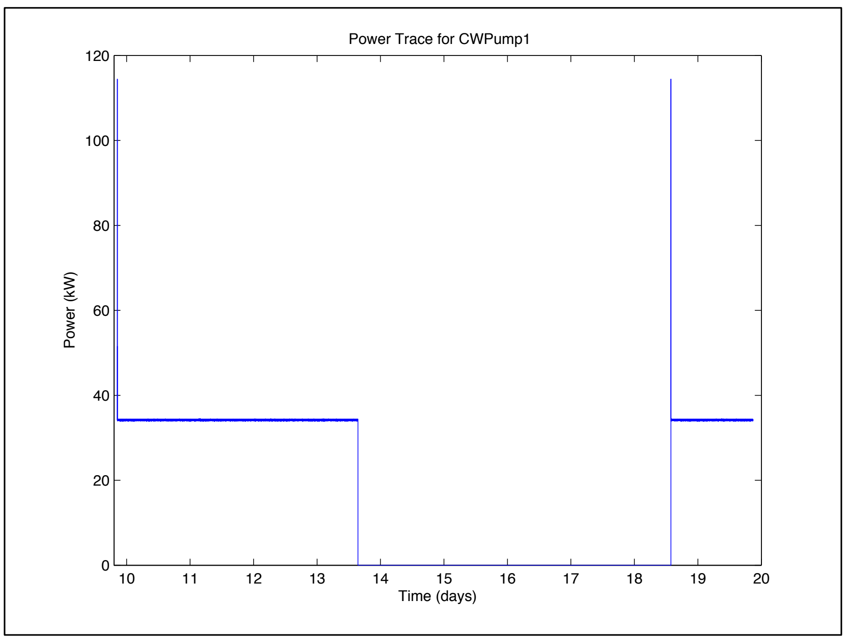

Figure 54: Model Results for AC Chill Water Pump #1

It is also of value to compare the total loading seen across all AC compressors to the loading profile seen in actual fleet data over the time frame. The results for this comparison are presented in Figure 55. These results show the strong correlation level between simulation and fleet data that can be created using the behavioral modeling.

The large spikes in the profile represent the operating condition where 4 compressors are on line in the intermediate state of switching AC plants. The simulation performs this randomly with similar periodicity to that is seen in the fleet, but it would not be expected to line up at corresponding times. The relative frequency and peak loading that occurs in this condition is also important when validating the results of a final model.

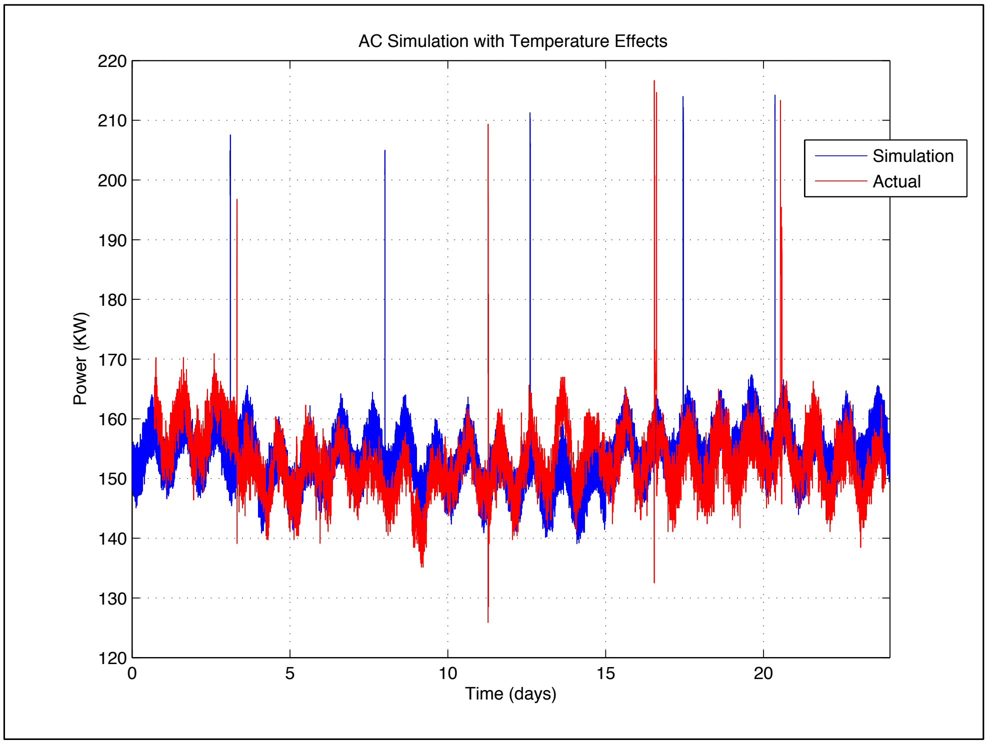

Figure 55: Comparison of AC Load Simulation and Actual Profile

The comparison between the simulated loading condition and the actual data seen in the fleet is again shown in Figure 56. From this comparison it can be seen that while the

simulation does not perfectly recreate the actual loading condition, it does experience changing load conditions similar to the actual data. For many of the dynamic questions now present for ship operations, the ability for the outputs to change with subtle changes in the input conditions would help ensure the proper technologies are pursued and implemented.

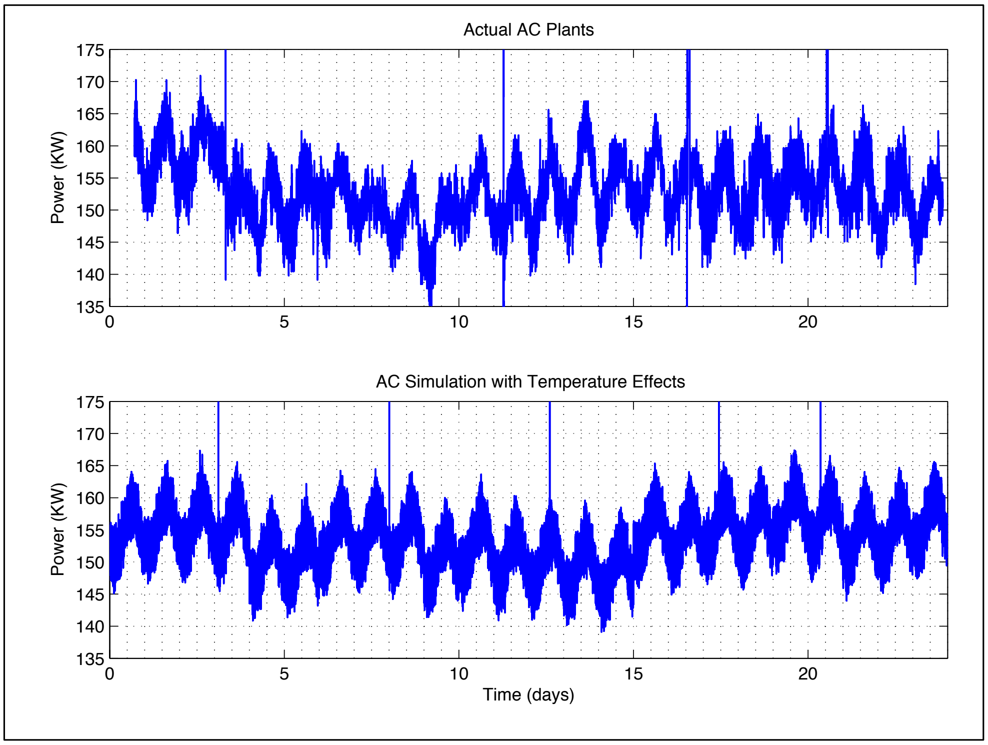

Figure 56: Comparison of Actual and Simulated AC Loads

5 Conclusions and Future Work

While this study of the EPLA methods in DDS 310-1 could be used as the basis of for improvement in many areas, a significant amount of work is still left in the field. Ideally, data could be gathered from relevant shipboard monitoring systems, such as MCMAS, and automatic updates could occur. Simple data mining tools could make updating operational profiles nearly effortless, and would present a means to improve upon methods discussed in this thesis.

5.1 Future Work

Utilizing available electronic data to update the information in DDS 310-1 is a topic with many avenues to be pursued. In the near-term, the data potentially available through the MCMAS system onboard ships presents a method of accurately updating the load factors used today. In addition to the load factors, they can provide the statistics necessary for accurately developing stochastic analyses or modeling systems. Working with fleet sponsors to create a statistically relevant data set across the fleet would be required to perform these tasks.

In the longer-term, developing a program that expanded the capability to model the ship in a simpler manner. Developing this program as a user-friendly program would move the modeling of the electric plant from one done on spreadsheets to one done through realistic modeling. As shown in the model developed here, this could present capability model things as they are known; simply modeling them with load factors when information is limited, and increasing fidelity as the design becomes clearer. This could allow a tool that is useful not only in sizing the electrical generating system, but could be useful throughout a ship's lifetime.

In developing the behavioral model, a significant amount of future work exists. Comparing the actual 1SA data to the simulated 1SA data (previously shown in Figure 50 and Figure 51), the degree of fluctuation about a mean operating state is not captured. One reason variation in the simulated data may be underrepresented is that there are weak dependencies in the electrical responses to system variables. By referring to the data

collected by Bennett at the LBES facility, the response of a main lube oil pump under maneuvering transients is shown in Figure 57. In this case, changes in the speed of the propulsion shafting causes change in lube oil system pressure, which causes slight changes in the operating point on the pump curve. Given the available data used for model development, these small changes could not be modeled. When summarized across many components, lack of fidelity could cause these errors. In future iterations of ship models, increasing the uncertainty in the individual loads by adjusting their PDF and CDF could improve the simulation characteristics.

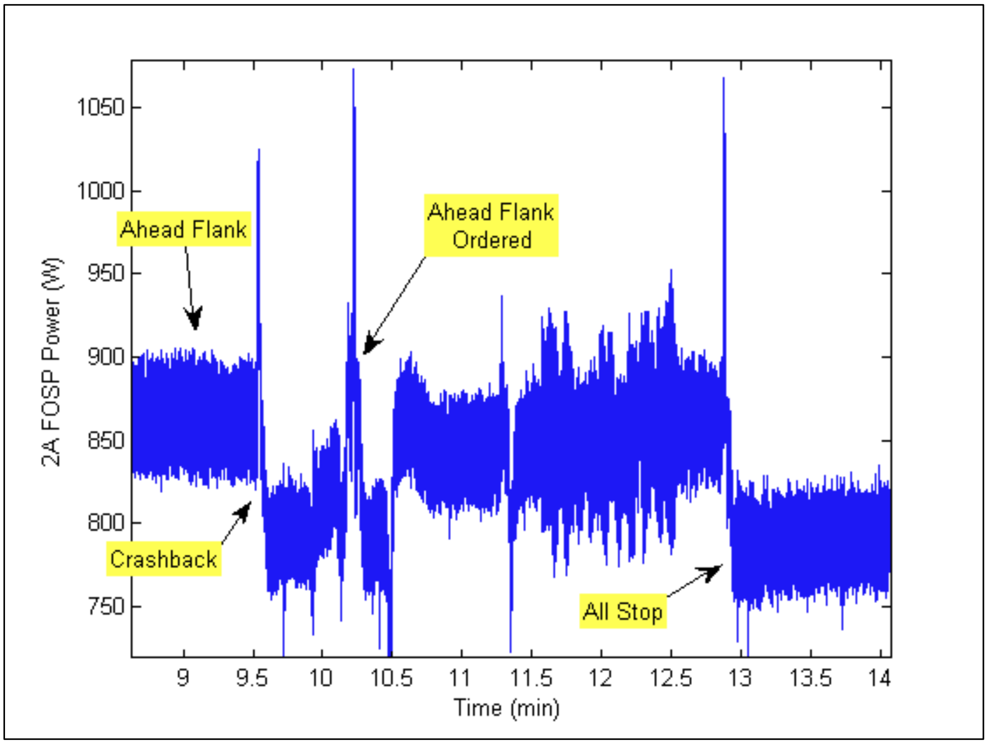

Figure 57: LBES Variation in MLO Pump With Maneuvering Transients [20]

Another potential cause for error in this model is some uncertainty in the data set used to develop the model. The data is a fairly comprehensive set of loads on the ship, but each load is individually only monitored for a few minutes in most cases. Monitoring for longer

periods, and under different operating conditions, could give an improved data series from which to draw from.

5.2 Conclusions

As US Navy ships continue their transition to the digital age an ever-increasing amount of information is available to those making design decisions. Current efforts utilize computer programs to perform tasks ranging from watchstander log taking to voyage planning. This thesis showed a couple of ways in which data availability could provide immediate impact in the design community. Continuation to other aspects of the electric plant design, and improvements in the quality of service areas are also fertile areas this research could be applied to.

The framework outlined for the development of a behavioral model is merely one method of implementing a solution for an increasingly complex problem. With a more diverse data set, the ability to simulate real-world shipboard responses to changing conditions could prove highly useful. By structuring a model such that the inputs and outputs are linked questions not yet formulated could be answered rapidly. This could be a benefit to the design community, operational planners, or anyone associated with Navy ships.

Table of Acronyms

AAW	Anti-Air Warfare
AC	Air Conditioning
ASUW	Anti-Surface Warfare
ASW	Anti-Submarine Warfare
CDF	Cumulative Distribution Function
CG	Guided Missile Cruiser
CPP	Controllable Pitch Propeller
DDG	Guided Missile Destroyer
DDS	Data Design Sheet
EPLA	Electrical Power Load Analysis
FOSP	Fuel Oil Service Pump
GTG	Gas Turbine Generator
GTM	Gas Turbine Module
HED	Hybrid Electric Drive
LBES	Land-Based Engineering Site
MCMAS	Machinery Control Message Acquisition System
MLO	Main Lubricating Oil
MRG	Main Reduction Gear
NAVSEA	Naval Sea Systems Command
NGIPS	Next Generation Integrated Propulsion Systems
NILM	Non-Invasive Load Monitoring
PDF	Probability Distribution Function
RMD	Restricted Maneuvering Doctrine
SHP	Shaft Horsepower
ZEDS	Zonal Electrical Distribution System

References

[1] J. S. Webster, H. Fireman, D. A. Allen, A. J. Mackenna, and J. C. Hootman, "Alternative propulsion methods for surface combatants and amphibious warfare ships," *Transactions- Society of Naval Architects and Marine Engineers*, vol. 115, pp. 224–262, Jan. 2007.

[2] N. Doerry, "Next Generation Integrated Power System (NGIPS) Technology Development Roadmap." NAVSEA, 30-Nov-2007.

[3] NAVSEA, "DDS 310-1: ELECTRIC POWER LOAD ANALYSIS (EPLA) FOR SURFACE SHIPS." Department of the Navy, Sep-2012.

[4] N. Doerry, "Electric Power Load Analysis," *Naval Engineers Journal*, vol. 124, no. 4, pp. 45–48, Dec. 2012.

[5] J. Amy and N. Doerry, "Implementing Quality of Service in Shipboard Power System Design," presented at the Electric Ship Technologies Symposium, 2011.

[6] Congressional Budget Office, "An Analysis of the Navy's Fiscal Year 2013 Shipbuilding Plan." Jul-2012.

[7] T. McCOY, J. Zgliczynski, N. W. Johanson, F. A. Puhn, and T. W. Martin, "Hybrid Electric Drive for DDG-51 Class Destroyers," *Naval Engineers Journal*, vol. 119, no. 2, pp. 83–91, 2007.

[8] D. Cusanelli and G. Karafiath, "Hydrodynamic Energy Saving Enhancements for DDG 51 Class Ships," ASNE Day 2012, Crystal City, VA, 09-Feb-2012.

[9] US NAVY, "United States Navy Fact File: Destroyers-DDG." Department of the Navy.

[10] US NAVY, "DDG 51 Hull number 91 and higher AC ZEDS training."

[11] NSWCCD-SSES, "A Guide to MFM III Operation." Naval Surface Warfare Center Carderock Division.

[12] I. R. McNab and F. C. Beach, "Naval Railguns," *IEEE Transactions on Magnetics*, vol. 43, no. 1, pp. 463–468, Jan.

[13] T. W. Webb, T. M. Kiehne, and S. T. Haag, "System-Level Thermal Management of Pulsed Loads on an All-Electric Ship," *IEEE Transactions on Magnetics*, vol. 43, no. 1, pp. 469–473, Jan.

[14] NAVSEA, "DDS 200-2: CALCULATION OF SURFACE SHIP ANNUAL ENERGY USAGE, ANNUAL ENERGY COST, AND FULLY BURDENED COST OF ENERGY." Department of the Navy, Aug-2012.

[15] S. Surko and M. Osborne, "Operating Speed Profles and the Ship Design Cycle," *Naval Engineers Journal*, vol. 117, no. 3, pp. 79–85, 2005.

[16] 05Z1 NAVSEA, "NAVSEA SPEED-TIME PROFILE GUIDE FOR SURFACE SHIPS." Department of the Navy, FY-2003.

[17] Team Ships, "PEO Ships Electric Ships Office." NAVSEA.

[18] B. Sievenpiper, T. Anderson, and K. Gerhard, "Operational Ship Utilization Modeling of the DDG-51 Class," presented at the ASNE DAY 2013, Crystal City, VA, 2013.

[19] J. A. Cairns, "DDG51 Class Land Based Engineering Site (LBES) – The Vision and the Value," *Naval Engineers Journal*, vol. 123, no. 2, pp. 73–83, 2011.

[20] P. Bennett, "Using the Non-Intrusive Load Monitor for Shipboard Supervisory Control," Master's Thesis, Massachusetts Institute of Technology, 2007.

[21] T. DeNucci, “Diagnostic Indicators for Shipboard Systems Using Non-Intrusive Load Monitoring,” Massachusetts Institute of Technology, 2005.
[22] A. Fuller, “Harmonic Approaches to Non-Intrusive Load Diagnostics,” Massachusetts Institute of Technology, 2008.
[23] D. Barber, “Shipboard Condition Based Maintenance and Integrated Power System Initiatives,” Massachusetts Institute of Technology, 2011.
[24] K. Douglas, “Shipboard Aggregate Power Monitoring,” Massachusetts Institute of Technology, 2009.
[25] W. Greene, “Evaluation of Non-Intrusive Monitoring for Condition Based Maintenance Applications on US Navy Propulsion Plants,” Massachusetts Institute of Technology, 2005.
[26] R. Jones, “Improving Shipboard Applications of Non-Intrusive Load Monitoring,” Massachusetts Institute of Technology, 2008.
[27] NAVSEA, “Ship Information Book DDG 100 USS KIDD, Part 1 Propulsion Plant.” NAVSEA, 28-Jul-2010.
[28] NAVSEA, “Ship Information Book DDG 100 USS KIDD, Part 2 Book 3 Auxiliary Machinery, Piping, and Envinronmental Pollution Control. Chapters 11-13.” NAVSEA, 28-Jul-2010.
[29] R. O’Rourke, “Navy DDG-51 and DDG-1000 Destroyer Programs: Background and Issues for Congress.” Congressional Research Service, 19-Apr-2011.
[30] Bath Iron Works, “DDG-51 Class Electric Load Analysis.” Department of the Navy.
[31] USCG Acquisition Directorate, “Maritime Security Cutter, Medium (WMSM).” US Coast Guard.
[32] T. Goodridge, “Baseline Report of the USS Spruance For Bath Iron Works.” Alaris Companies, Nov-2010.

Appendix A: MATLAB Code Developed For Simulation

This appendix contains the primary functional portions of the MATLAB code developed for the simulation of the 1SA switchboard. Uzoma Orji wrote the code referenced in this appendix as part of the collaborative efforts to develop these simulation techniques.

Appendix A.1: Ship System.m

```
classdef ShipSystem < handle
    properties
        Name
        Subsystems = {};
        SelectionMethod = 'random-subset';
        RandomSubsetParam
        SequentialList
        LevelRulesText
        LevelRules
        StateModel
        OnModel
        OffModel
        TimeDependencies
        FSMCurrentState = 1;
        FSMTransitions
        FSMModels
        SubsystemOnOffVectors
    end % properties

    methods
        function on_times = initializeOnOffVectors(self)
            OnOffVectors = cell(1,numel(self.Subsystems));
            on_times = cell(1,numel(self.Subsystems));

            for k = 1:numel(self.Subsystems)
                OnOffVectors{k} = cell(1,numel(self.Subsystems{k}));
                on_times{k} = cell(1,numel(self.Subsystems{k}));
            end
            self.SubsystemOnOffVectors = OnOffVectors;
        end % initializeOnOffVectors

        function runSubsystems(self,TimeLength, TimeStep, TimeStart)
            on_times = self.initializeOnOffVectors();

            t = TimeStart;
            TimeEnd = TimeStart + TimeLength;

            while ( t < TimeEnd )
                if strcmpi(self.SelectionMethod,'single-state')
                    for k = 1:numel(self.Subsystems)
                        on_times{k}{1} = [TimeStart TimeEnd];
                    end
                    t = TimeEnd + TimeStep;
                elseif strcmpi(self.SelectionMethod,'cycle')
                    t_rand = getRandomModelValue(self.OffModel);
```

```
                    t = t + t_rand;
                    t_rand = getRandomModelValue(self.OnModel);
                    on_times{1}{1} = [on_times{1}{1}; [t, t+t_rand]];
                    t = t + t_rand;
        elseif strcmpi(self.SelectionMethod,'fsm')
                    tempfsmmodel = self.FSMModels{self.FSMCurrentState};
                    t_rand = getRandomModelValue(tempfsmmodel);
                    tempstate = self.FSMTransitions(self.FSMCurrentState,:);
                    rnum = rand(1);
                    probs = cumsum(tempstate{1})>rnum;
                    nextstate = find(probs==1,1);
                    k = self.FSMCurrentState;
                    n = randsample(numel(self.Subsystems{k}), 1);
                    on_times{k}{n} = [on_times{k}{n}; [t, t+t_rand]];
                    self.FSMCurrentState = nextstate;
                    t = t + t_rand;
        elseif strcmpi(self.SelectionMethod,'level')
                    for k = 1:numel(self.Subsystems)
                        data =
evaluateRule(self.LevelRules{k},TimeLength,TimeStep,1);
                        on_times{k}{1} =
getOnTimes(data,TimeStart:TimeStep:TimeEnd);
                    end
                    t = TimeEnd + TimeStep;
                elseif strcmpi(self.SelectionMethod,'time-dependency')
                    for k = 1:numel(self.Subsystems)
                        on_times{k}{1} =
getTimeDependency(self.TimeDependencies{k}, TimeLength, TimeStep);
                    end
                    t = TimeEnd + TimeStep;
                elseif strcmpi(self.SelectionMethod,'random-subset')
                    t_rand = getRandomModelValue(self.StateModel);
                    arr = randsample(numel(self.Subsystems),
self.RandomSubsetParam);
                    arr = sort(arr);
                    for ind = 1:numel(arr)
                        k = arr(ind);
                        n = randsample(numel(self.Subsystems{k}), 1);
                        on_times{k}{n} = [on_times{k}{n}; [t, t+t_rand]];
                    end
                    t = t + t_rand;
                else
                    error('SelectionMethod has an unknown value');
                end
            end % while

            for k = 1:numel(on_times)
                for l = 1:numel(on_times{k})
                    onoffvec = zeros(round(TimeLength/TimeStep)+1,1);
                    cur_on_times = on_times{k}{l};
                    N = size(cur_on_times, 1);
                    for n = 1:N
                        ind_start = round((cur_on_times(n,1)-
TimeStart)/TimeStep)+1;
                        ind_end = cur_on_times(n,2);
                        if (ind_end > TimeEnd)
                            ind_end = TimeEnd;
                        end
```

```
                        ind_end = round((ind_end-TimeStart)/TimeStep)+1;
                        onoffvec(ind_start:ind_end) = 1;
                    end
                    self.SubsystemOnOffVectors{k}{l} = onoffvec;
                end % for
            end % for
        end % runSubsystems
    end % methods
end %classdef
```

Appendix A.2: ShipComponent.m

```
classdef ShipComponent < handle
    properties
        Name
        Type = 'slave';
        OnModel
        OffModel
        OnOffset = 0;
        OffOffset = 0;
        LoadCenter
        SystemNegate = false;
        on_times
        OnOffVector
        PowerTraceType = 'fingerprint'
        LoadPower
        LoadFactor
        SimLoadFactor
        TransientTurnOn
        SteadyState
        TransientTurnOff
        PowerTimeVars
        TemperatureOffset
        TemperaturesVarName
        Temperatures
        PowerFSM
        PowerTrace
    end % properties

    methods
          function run(self,TimeLength,TimeStep,TimeStart,SystemOnOffVector)
            if ( self.SystemNegate )
                SystemOnOffVector = ~SystemOnOffVector;
            end

            shiftedleft = SystemOnOffVector;
            shiftedright = SystemOnOffVector;

            if ( self.OnOffset ~= 0 )
                Noffset = round(self.OnOffset/TimeStep);
                shiftedleft = [SystemOnOffVector(Noffset+1:end);
zeros(Noffset,1)];
            end

            if ( self.OffOffset ~= 0 )
                Noffset = round(self.OffOffset/TimeStep);
```

```
                shiftedright = [zeros(Noffset,1); SystemOnOffVector(1:end-
Noffset)];
            end

            SystemOnOffVector = shiftedleft | shiftedright;

            TimeEnd = TimeStart+TimeLength;

            switch lower(self.Type)
                case 'slave'
                    self.OnOffVector = SystemOnOffVector;
                    self.on_times =
getOnTimes(SystemOnOffVector,TimeStart:TimeStep:TimeEnd);
                case 'master'
                    on_times =
getOnTimes(SystemOnOffVector,TimeStart:TimeStep:TimeEnd);
                    begin_times = on_times(:,1)';
                    end_times = on_times(:,2)';

                    Nruns = numel(begin_times);
                    on_times = [];

                    for n = 1:Nruns
                        t = begin_times(n);
                        temp_end_time = end_times(n);
                        while ( t < temp_end_time )
                            % compute the duration of the current off time
                            cur_Model = getCurrentModel(self.OffModel);
                            t_rand = getRandomModelValue(cur_Model);
                            seg_start = t + t_rand;

                            % compute the duration of the current on time
                            cur_Model = getCurrentModel(self.OnModel);
                            t_rand = getRandomModelValue(cur_Model);
                            seg_end = min(seg_start + t_rand, temp_end_time);

                            on_times = [on_times; [seg_start seg_end]];
                            t = seg_end;
                        end % while
                    end % for

                    self.on_times = on_times;

                    N = size(on_times, 1);

                    onoffvec = zeros(round(TimeLength/TimeStep)+1,1);

                    for n = 1:N
                        ind_start = round((on_times(n,1)-
TimeStart)/TimeStep)+1;
                        ind_end = on_times(n,2);
                        if (ind_end > TimeEnd)
                            ind_end = TimeEnd;
                        end
                        ind_end = round((ind_end-TimeStart)/TimeStep)+1;
                        onoffvec(ind_start:ind_end) = 1;
```

```
                    end
                    self.OnOffVector = onoffvec;
                otherwise
                    error('Type has an unknown value');
            end % switch
        end % run

        function getPowerTrace(self, TimeLength, TimeStep, TimeStart)
            self.PowerTrace = zeros(round(TimeLength/TimeStep)+1,1);
            N = size(self.on_times, 1);
            for n = 1:N
                ind_start = round((self.on_times(n,1)-TimeStart)/TimeStep)+1;
                ind_end = round((self.on_times(n,2)-TimeStart)/TimeStep)+1;
                int_len = ind_end-ind_start+1;

                if (strcmpi(self.PowerTraceType,'constant'))
                    self.PowerTrace(ind_start:ind_end) =
self.LoadPower*self.LoadFactor;
                elseif (strcmpi(self.PowerTraceType,'time-dependency'))
                    self.PowerTrace(ind_start:ind_end) =
getTimeDependentPowerTrace(self.PowerTimeVars,int_len,self.on_times(n,1));
                elseif (strcmpi(self.PowerTraceType,'fsm'))
                    self.PowerTrace(ind_start:ind_end) =
getFSMPowerTrace(self.PowerFSM,int_len);
                elseif (strcmpi(self.PowerTraceType,'fingerprint'))
                    if ~isempty(self.TransientTurnOn)
                        if (strcmpi(self.TransientTurnOn{1},'fixed-model'))
                            TempTurnOn = zeros(self.TransientTurnOn{2},1);
                            for k = 1:self.TransientTurnOn{2}
                                TempTurnOn(k) =
getRandomModelValue(self.TransientTurnOn{3});
                            end
                        else
                            TempTurnOn = self.TransientTurnOn{2};
                        end
                        lenOn = numel(TempTurnOn);
                        lenOff = numel(self.TransientTurnOff{2});

                        if (int_len < lenOff)
                            self.PowerTrace(ind_start:ind_end) =
self.TransientTurnOff{2}(end-int_len+1:end);
                        elseif (int_len < lenOn + lenOff)
                            self.PowerTrace(ind_start:ind_end) =
[getSteadyStateVector(self.SteadyState, int_len-lenOff);
self.TransientTurnOff{2}];
                        else
                            self.PowerTrace(ind_start:ind_start+lenOn-1) =
TempTurnOn;
                            self.PowerTrace(ind_start+lenOn:ind_end-lenOff) =
getSteadyStateVector(self.SteadyState, int_len-lenOn-lenOff);
                            self.PowerTrace(ind_end-lenOff+1:ind_end) =
self.TransientTurnOff{2};
                        end % if
                    end % if
                end % if
                if ~isempty(self.TemperatureOffset)
                    self.PowerTrace =
addTemperatureOffset(self.PowerTrace,self.PowerTimeVars,int_len,self.on_times
```

```
(n,1),self.Temperatures,self.TemperatureOffset);
                end
            end % for
        end % getPowerTrace

        function getSimLoadFactor(self, TimeLength, TimeStep)
            self.SimLoadFactor =
sum(self.PowerTrace)/TimeLength/self.LoadPower;
        end % getSimLoadFactor
    end % methods
end %classdef
```

Appendix A.3: Time Dependent Power Traces

```
function PowerVector = getTimeDependentPowerTrace(PowerTimeVars,VectorLength,
tstart)
% PowerVector = getTimeDependentPowerTrace(PowerTimeVars,VectorLength)
%    This function breaks the day into 12 2-hr time blocks
%    PowerTimeVars has a 12 element array to describe the mean and
%    standard deviation for each time block.

t = tstart:tstart+VectorLength-1;
day = t/(3600*24);
day_rem = day - floor(day);
hr = floor(day_rem*24);
block_ind = floor(hr/2)+1;

mu = PowerTimeVars{1}(block_ind);
std = PowerTimeVars{2}(block_ind);

PowerVector = randn(1,VectorLength).*std+mu;
```

Appendix A.4: Code to Create Time Dependency

```
function on_times = getTimeDependency(t_arr, TimeLength, TimeStep, TimeStart)
%getTimeDependency Get the start and end times of on runs dependent on t_arr
%   on_times = getOnTimes(t_arr, TimeLength, TimeStep, TimeStart)
%          t_arr : array of time intervals
%     TimeLength : simulation parameter
%       TimeStep : simulation parameter
%      TimeStart : simulation parameter

t = TimeStart:TimeStep:TimeStart+TimeLength;

t_display = datenum(datestr(t/86400,13));

onoffvector = zeros(size(t_display));

for k = 1:numel(t_arr)
    start_time = datenum(t_arr{k}{1});
    end_time = datenum(t_arr{k}{2});

    onoffvector = onoffvector | ((t_display >= start_time) & (t_display <=
end_time));
end
```

```
on_times = getOnTimes(onoffvector,t);
end
```

Appendix A.5: Steady State Vector

```
function SSVector = getSteadyStateVector(SSModel,VectorLength)
% SSVector = getSteadyStateVector(SSModel,VectorLength)
SSVector = zeros(VectorLength,1);

SStype = SSModel{1};

if (strcmpi(SStype,'fingerprint'))
    lenSS = numel(SSModel{2});

    if (lenSS > VectorLength)
        SSVector = SSModel{2}(1:VectorLength);
    else
        N_SS = floor(VectorLength/lenSS);
        mod_SS = mod(VectorLength,lenSS);
        tempstart = 1;
        for k = 1:N_SS
            SSVector(tempstart:tempstart+lenSS-1) = SSModel{2};
            tempstart = tempstart + lenSS;
        end
        SSVector(tempstart:tempstart+mod_SS-1) = SSModel{2}(1:mod_SS);
    end
elseif (strcmpi(SStype,'multiple'))
    ind = 1;
    for k = 1:VectorLength
        rnum = rand(1);
        probs = cumsum(SSModel{2}{ind})>rnum;
        ind = find(probs==1,1);
        SSVector(k) = getRandomModelValue(SSModel{3}{ind});
    end
elseif (strcmpi(SStype,'normal'))
    SSVector = getNormalSteadyState(SSModel{2},SSModel{3},VectorLength);
else
    for k = 1:VectorLength
        SSVector(k) = getRandomModelValue(SSModel);
    end
end
```

Appendix A.6: Stochastic Model Generation

```
function val = getRandomModelValue(Model)
% val = getRandomModelValue(Model)
% Possible Models:
%          {'deterministic',a}
%            a -> value of constant
%          {'uniform',a,b}
%           a -> left endpoint of interval
%           b -> right endpoint
%          {'exponential',lambda}
%            lambda -> arrival rate
%          {'poisson',lambda}
```

```
%           lambda -> arrival rate

val = -1;

while val < 0
    if ( strcmpi(Model{1}, 'determinisitc') )
        val = Model{2};
    elseif ( strcmpi(Model{1}, 'uniform') )
        val = rand(1);
        val = val*(Model{3}-Model{2}) + Model{2};
    elseif ( strcmpi(Model{1}, 'normal') )
        val = randn(1,1);
        val = val*Model{3}+Model{2};
    elseif ( strcmpi(Model{1}, 'exponential') )
        val = rand(1);
        val = expinv(val,1/Model{2});
    elseif ( strcmpi(Model{1}, 'poisson') )
        val = rand(1);
        val = poissinv(val,Model{2});
    else
        error('Model has an unknown type');
    end % if
end
```

Appendix A.7: Creates Vectors of Equipment Start Times

```
function on_times = getOnTimes(data, t)
%getOnTimes Get the start and end times of on runs
%   on_times = getOnTimes(data, t)
%    data : the row vector containing the on/off usage of load
%       t : corresponding time vector
% on_times: Nx2 array of start and end times of on runs

    ind_ons = find(data == 1);

    on_times = [];
    if (~isempty(ind_ons))
        indend = numel(ind_ons);
        curind = 2;
        temp_on_times = [t(ind_ons(1)) 0];

        while (curind <= indend)
            if (ind_ons(curind) ~= ind_ons(curind-1) + 1)
                temp_on_times(2) = t(ind_ons(curind-1));
                on_times = [on_times; temp_on_times];
                temp_on_times(1) = t(ind_ons(curind));
            end
            curind = curind+1;
        end
        on_times = [on_times; [temp_on_times(1) t(ind_ons(end))]];
    end
end
```

Appendix A.8: Normal Steady State Power Model

```
function SSVector = getNormalSteadyState(mu,std,VectorLength)
% SSVector = getNormalSteadyState(mu,std,VectorLength)

SSVector = (randn(VectorLength,1)*std)+mu;
inds = find(SSVector<0);
for ind = inds;
    SSVector(ind) = getRandomModelValue({'normal',mu,std});
end
```

Appendix A.9: FSM Power Trace Generation

```
function PowerVector = getFSMPowerTrace(PowerFSM,VectorLength)
% PowerVector = getFSMPowerTrace(PowerFSM,VectorLength)

%POWERFSM = { {1, {'normal',2,3}, {'normal',9.25,0.5}, .05} };

PowerVector = zeros(1,VectorLength);
remLen = VectorLength;
curstate = 1;
while remLen > 0
    rnum = rand(1);
    tempstate = PowerFSM{curstate};
    tempLength = floor(getRandomModelValue(tempstate{2}));
    tempLength = min(tempLength,remLen);

    mu = getRandomModelValue(tempstate{3});
    std = tempstate{4};

    PowerVector(VectorLength-remLen+1:VectorLength-remLen+tempLength) =
getNormalSteadyState(mu,std,tempLength);

    probs = cumsum(tempstate{1})>rnum;
    curstate = find(probs==1,1);
    remLen = remLen - tempLength;
end
```

Appendix A.10: Evaluate Model Rules

```
function data = evaluateRule(Rules, TimeLength, TimeStep, level)
% data = evaluateRule(Rules, TimeLength, TimeStep, level)
% Examples of Rule Structure:
%  (x > 0) -> {{{x,'gt',0}}}
%  (x < 0) & (y == 1) -> {{{x,'gt',0}}{{y,'eq',1}}}
%  (x > 0) || (x < 0) & (y == 1) -> {{{x,'gt',0}}{{x,'gt',0}}{{y,'eq',1}}}

if (level == 1)
    data = zeros((round(TimeLength/TimeStep)+1),1);
else
    data = ones((round(TimeLength/TimeStep)+1),1);
end

if (level == 1)
```

```
    for k = 1:numel(Rules)
        data = data | evaluateRule(Rules{k}, TimeLength, TimeStep, 2);
    end
elseif (level == 2)
    for k = 1:numel(Rules)
        data = data & evaluateRule(Rules{k}, TimeLength, TimeStep, 3);
    end
else
    op = Rules{2};
    var = Rules{1};
    num = Rules{3};

    var = var(1:round(TimeLength/TimeStep)+1);

    if ( strcmpi(op, 'eq') )
        data = data & ( var == num );
    elseif ( strcmpi(op, 'lt') )
        data = data & ( var < num );
    elseif ( strcmpi(op, 'gt') )
        data = data & ( var > num );
    elseif ( strcmpi(op, 'leq') )
        data = data & ( var <= num );
    elseif ( strcmpi(op, 'geq') )
        data = data & ( var >= num );
    else
        error('Rule has an unknown operation');
    end % if
end
```

Appendix A.11: Check Model Parameters

```
  function isModel = checkModel(Model)
  ModelElements = numel(Model);
  if (ModelElements <= 1)
      error('Model must have more than 1 element');
  end % if

  ModelName = Model{1};

  if ( strcmpi(ModelName, 'uniform') )
      if (ModelElements ~= 3)
          error('Model has an incorrect number of parameters');
      end
      if (Model{3} < Model{2})
          error('Model parameters are not possible');
      end
  elseif ( strcmpi(ModelName, 'normal') )
      if (ModelElements ~= 3)
          error('Model has an incorrect number of parameters');
      end
  elseif ( strcmpi(ModelName, 'deterministic') || strcmpi(ModelName,
'exponential') || strcmpi(ModelName, 'poisson') )
      if (ModelElements ~= 2)
          error('Model has an incorrect number of parameters');
      end
  else
      error('Model has an unknown value');
```

```
  end % if
  isModel = true;
  end % checkModel
```

Appendix A.12: Main GUI Program

```
function varargout = mainGUI

hf = findall(0,'Tag',mfilename);
if ~isempty(hf)
    close(hf);
end

hf = localCreateUI;

ad = guidata(hf);

% populate the output if required
if nargout > 0
%     varargout{1} = hf;
    varargout{1} = ad;
end


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Function to create the user interface
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function hf = localCreateUI
%try
    % Create the figure, setting appropriate properties
    hf = figure('Tag',mfilename,...
        'Toolbar','none',...
        'MenuBar','none',...
        'IntegerHandle','off',...
        'Position', [624 196 1080 665],...
        'Units','normalized',...
        'Resize','off',...
        'NumberTitle','off',...
        'HandleVisibility','callback',...
        'Name','Ship Design Simulation',...
        'CloseRequestFcn',@localCloseRequestFcn,...
        'Visible','off');

    hsp = uipanel('Parent',hf,...
        'Units','pixels',...
        'Position', [12 34 595 620],...
        'Title','Simulation',...
        'BackgroundColor',get(hf,'Color'),...
        'HandleVisibility','callback',...
        'Tag','simPanel');


    btnTexts = {'Edit Global Inputs', 'Edit Ship Description', ...
        'Edit Simulation Parameters'};
    btnY = [460 181 65];
    for idx = 1:3
```

```
    uicontrol('Parent',hsp,...
        'Style','pushbutton',...
        'Units','pixels',...
        'Position',[7 btnY(idx) 210 28],...
        'BackgroundColor',get(hf,'Color'),...
        'String',btnTexts{idx},...
        'Callback',@localSimButtonPressed,...
        'HandleVisibility','callback',...
        'Tag',sprintf('SimBtn%d',idx));
end


panelTexts = {'Global Inputs', 'Ship Description', ...
    'Simulation Parameters'};
panelH = [134 264 92];
textH = [106 236 64];
for idx = 1:3
    hp1 = uipanel('Parent',hsp,...
        'Units','pixels',...
        'Position', [233 btnY(idx) 351 panelH(idx)],...
        'Title',panelTexts{idx},...
        'BackgroundColor',get(hf,'Color'),...
        'HandleVisibility','callback',...
        'Tag',sprintf('simPanel%d',idx));

    ht1 = uicontrol('Parent',hp1,...
        'Style','text',...
        'Units', 'pixels',...
        'Position',[7 5 334 textH(idx)],...
        'BackgroundColor',get(hf,'Color'),...
        'HorizontalAlignment','left',...
        'ForegroundColor',[0 0 0],...
        'HandleVisibility','callback',...
        'Tag',sprintf('text%d',idx));
end

hsb = uicontrol('Parent',hsp,...
    'Style','pushbutton',...
    'Units','pixels',...
    'Position', [212 12 139 33],...
    'BackgroundColor',get(hf,'Color'),...
    'String','Run Simulation',...
    'Callback',@localRunPressed,...
    'HandleVisibility','callback',...
    'Tag','buttonrun');

hop = uipanel('Parent',hf,...
    'Units','pixels',...
    'Position', [649 165 351 372],...
    'Title','Outputs',...
    'BackgroundColor',get(hf,'Color'),...
    'HandleVisibility','callback',...
    'Tag','outPanel');

btnTexts = {'Get Power Traces', 'Get Load Factors',...
    'Calculate Fuel Consumption', 'Other Stuff'};
btnY = [0.8 0.6 0.4 0.2];
```

```
    for idx = 1:4
        uicontrol('Parent',hop,...
            'Style','pushbutton',...
            'Units','normalized',...
            'Position',[0.2 btnY(idx) 0.6 0.10],...
            'BackgroundColor',get(hf,'Color'),...
            'ForegroundColor',[0 0 0],...
            'String',btnTexts{idx},...
            'Callback',@localOutputButtonPressed,...
            'HandleVisibility','callback',...
            'Enable','off',...
            'Tag',sprintf('OutBtn%d',idx));
    end


    % Simulation Parameters
    ad.output = hf;
    ad.handles = guihandles(hf);

    ad = loadPreset(ad);
%     ad.TimeStep = 1;
%     ad.TimeLength = 287999;
%     ad.TimeStart = 0;
%     ad.Speed = 'speedprofile.dat';
%     ad.Temperature = 'temperatureprofile.dat';
%     ad.Engine = 'engineprofile.dat';
%     ad.Season = 'Summer';
%     ad.AllSystems = {};



    str = sprintf('Speed Profile: %s\nTemperature Profile: %s\nEngine
Profile: %s\nSeason: %s',...
        ad.Speed,ad.Temperature,ad.Engine,ad.Season);
    set(ad.handles.text1,'String',str);

    str = sprintf('Time Step: %d\nTime Length: %d\nTime Start:
%d',ad.TimeStep,ad.TimeLength,ad.TimeStart);
    set(ad.handles.text3,'String',str);

    str = '';

    for idw = 1:numel(ad.AllSystems)
        str = [str sprintf('%s\n  ',ad.AllSystems{idw}(4:end))];
        tempSystem = eval(ad.AllSystems{idw});
        for idx = 1:numel(tempSystem.Subsystems)
            tempLevel1 = tempSystem.Subsystems{idx};
            for idy = 1:numel(tempLevel1)
                str = [str '['];
                tempLevel2 = tempLevel1{idy};
                for idz = 1:numel(tempLevel2)-1
                    str = [str sprintf('%s, ',tempLevel2{idz}(4:end))];
                end
                str = [str sprintf('%s] ',tempLevel2{end}(4:end))];
            end
            str = [str '\n  '];
```

```
        end
       str = [str '\n'];
    end
    str = sprintf(str);
    set(ad.handles.text2,'String',str);

    guidata(hf, ad);
    % Position the UI in the centre of the screen
    movegui(hf,'center')
    % Make the UI visible
    set(hf,'Visible','on');
% catch ME
%     % Get rid of the figure if it was created
%     if exist('hf','var') && ~isempty(hf) && ishandle(hf)
%         delete(hf);
%     end
%     % throw up an error dialog
%     estr = sprintf('%s\n%s\n\n',...
%         'The UI could not be created.',...
%         'The specific error was:',...
%         ME.message);
%     errordlg(estr,'UI creation error','modal');
% end


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Callback Function for Simulation Button
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function localSimButtonPressed(hObject,eventdata)

ad = guidata(hObject);

switch get(hObject,'Tag')
    case 'SimBtn1'
        loadGlobalInputs(ad.output);
    case 'SimBtn2'
        loadShipSystems(ad.output);
    case 'SimBtn3'
        loadSimulationParams(ad.output);
    otherwise % shouldn't be able to get in here
        errordlg('Selection Error', 'modal');
end


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Callback Function for Run button
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function localRunPressed(hObject,eventdata)

% get the application data
ad = guidata(hObject);

TimeStep = ad.TimeStep;
TimeLength = ad.TimeLength;
TimeStart = ad.TimeStart;
AllSystems = ad.AllSystems;
```

```
powertrace = zeros(round(TimeLength/TimeStep)+1,1);
numloads = 0;
loadsim = 0;
for k = 1:numel(AllSystems)
    SS = eval(AllSystems{k});
    for l = 1:numel(SS.Subsystems)
        for m = 1:numel(SS.Subsystems{l})
            numloads = numloads + numel(SS.Subsystems{l}{m});
        end
    end
end

H = waitbar(0,'Starting simulation...');

for k = 1:numel(AllSystems)
    SS = eval(AllSystems{k});
    waitbar(0,H,sprintf('Simulating System: %s',SS.Name));
    SS.runSubsystems(TimeLength,TimeStep,TimeStart);
    for l = 1:numel(SS.Subsystems)
        for m = 1:numel(SS.Subsystems{l})
            OnOffVector = SS.SubsystemOnOffVectors{l}{m};
            for n = 1:numel(SS.Subsystems{l}{m})
                tempload = eval(SS.Subsystems{l}{m}{n});
                loadsim = loadsim + 1;
                waitbar(loadsim/numloads,H,sprintf('In System: %s\n
Simulating Load %s',SS.Name,tempload.Name));
                tempload.run(TimeLength,TimeStep,TimeStart,OnOffVector);
                if ~isempty(tempload.LoadCenter)
                    if (tempload.LoadCenter==11) || (tempload.LoadCenter==23)
|| (tempload.LoadCenter==61)
                        tempload.getPowerTrace(TimeLength, TimeStep,
TimeStart);
                        powertrace = powertrace + tempload.PowerTrace;
                        tempload.getSimLoadFactor(TimeLength, TimeStep);
                    end
                end
            end
        end
    end
end

delete(H);
set(findobj('Tag','OutBtn1'),'Enable','On');
set(findobj('Tag','OutBtn2'),'Enable','On');

ad.powertrace = powertrace;
guidata(hObject,ad);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Callback Function for Output Button
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function localOutputButtonPressed(hObject,eventdata)

ad = guidata(hObject);

switch get(hObject,'Tag')
```

```
    case 'OutBtn1'
        loadPowerTraces(ad.output);
    case 'OutBtn2'
        loadLoadFactors(ad.output);
    case 'OutBtn3'
    case 'OutBtn4'
    otherwise % shouldn't be able to get in here
        errordlg('Selection Error', 'modal');
end


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Callback Function for deleting the UI
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function localCloseRequestFcn(hObject,eventdata)

% destroy the window
delete(gcbo);


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Load Engine Profile
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function ad = loadEngineProfile(old_ad)
ad = old_ad;

fid = fopen(ad.Engine);
while 1
    tline = fgetl(fid);

    if ~ischar(tline)
        break;
    end

    words = mystrip(tline);

    if (numel(words)>1)
        val = words{2};
    end

    VarName = words{1};

    if strcmp(VarName, '};')
        break;
    elseif strcmp(val, '{')
    else
        eval(sprintf('ad.%s = load(''%s'');',VarName,val));
    end
end
fclose(fid);


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Load Preset
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function ad = loadPreset(old_ad)
```

```
fid = fopen('guipreset.abc');

inShipSystem = false;
inSubsystems = false;
inShipLoad = false;
inGroup = false;
inLevelRulesText = false;
inLevel1 = false;
inLevel2 = false;
inLevelAnd = false;
inLevelOr = false;
inLevelRule = false;
inLevelRulesText = false;
tempSystem = ShipSystem;
tempSubsystems = {};
tempLoad = ShipLoad;

myQueue = {};

tempLevel1 = {};
tempLevel2 = {};
tempLevelAnd = {};
tempLevelOr = {};
tempLevelRule = {};
tempLevelRulesText = {};

ad = old_ad;
ad.AllSystems = {};

while 1
    tline = fgetl(fid);

    if ~ischar(tline)
        break;
    end

    words = mystrip(tline);

    if (numel(words)>1)
        if strcmp(words{2},'{')
            val = words{2};
        else
            val = eval(words{2});
        end
    end

    PropName = words{1};
    switch PropName
        case 'Simulation'
        case 'Name'
            if (inShipLoad)
                tempLoad.Name = val;
            elseif (inShipSystem)
                tempSystem.Name = val;
            end
        case 'Speed'
```

```
    ad.Speed = val;
    eval(sprintf('ad.SpeedData = load(''%s'');',val));
case 'Temperature'
    ad.Temperature = val;
    eval(sprintf('ad.Temperatures = load(''%s'');',val));
case 'Engine'
    ad.Engine = val;
    ad = loadEngineProfile(ad);
case 'Season'
    ad.Season = val;
case 'TimeStep'
    ad.TimeStep = val;
case 'TimeLength'
    ad.TimeLength = val;
case 'TimeStart'
    ad.TimeStart = val;
case 'ShipSystem'
    inShipSystem = true;
    tempSystem = ShipSystem;
    myQueue = { 'ShipSystem', myQueue{1:end} };
case 'Subsystems'
    inSubsystems = true;
    myQueue = { 'Subsystems', myQueue{1:end} };
case 'ShipLoad'
    inShipLoad = true;
    tempLoad = ShipLoad;
    myQueue = { 'ShipLoad', myQueue{1:end} };
case 'Level1'
    inLevel1 = true;
    myQueue = { 'Level1', myQueue{1:end} };
case 'Level2'
    inLevel2 = true;
    myQueue = { 'Level2', myQueue{1:end} };
case 'Cell'
    tline2 = fgetl(fid);
    cellwords = mystrip(tline2);
    dim = eval(cellwords{2});
    tempCell = cell(1,dim);
    for k = 1:dim
        tline2 = fgetl(fid);
        cellwords = mystrip(tline2);
        element = eval(cellwords{2});
        tempCell{k} = element;
    end
    myQueue = { 'Cell', myQueue{1:end} };
case 'TemperaturesVarName'
    tempLoad.TemperaturesVarName = val;
    eval(sprintf('tempLoad.Temperatures = %s;',val));
case 'LevelAnd'
    inLevelAnd = true;
    myQueue = { 'LevelAnd', myQueue{1:end} };
case 'LevelOr'
    inLevelOr = true;
    myQueue = { 'LevelOr', myQueue{1:end} };
case 'LevelRule'
    inLevelRule = true;
    myQueue = { 'LevelRule', myQueue{1:end} };
case '}'
```

```
            celltype = myQueue{1};
            switch celltype
                case 'ShipSystem'
                    eval(sprintf('ad.%s =
carboncopy(tempSystem);',tempSystem.Name));
                    eval(sprintf('ad.AllSystems = { ad.AllSystems{1:end},
''ad.%s'' };',tempSystem.Name));
                    inShipSystem = false;
                case 'Subsystems'
                    tempSystem.Subsystems = tempSubsystems;
                    inSubsystems = false;
                    tempSubsystems = {};
                case 'ShipLoad'
                    eval(sprintf('ad.%s =
carboncopy(tempLoad);',tempLoad.Name));
                    eval(sprintf('tempLevel2 = { tempLevel2{1:end}, ''ad.%s''
};',tempLoad.Name));
                    inShipLoad = false;
                case 'Level2'
                    tempLevel1 = { tempLevel1{1:end}, tempLevel2 };
                    inLevel2 = false;
                    tempLevel2 = {};
                case 'Level1'
                    tempSubsystems = { tempSubsystems{1:end}, tempLevel1 };
                    inLevel1 = false;
                    tempLevel1 = {};
                case 'LevelAnd'
                    tempLevelOr = { tempLevelOr{1:end}, tempLevelAnd };
                    inLevelAnd = false;
                    tempLevelAnd = {};
                case 'LevelOr'
                    % tempLevelRule = { tempLevelRule{1:end}, tempLevelOr };
                    tempLevelRulesText = { tempLevelRulesText{1:end},
tempLevelOr };
                    inLevelOr = false;
                    tempLevelOr = {};
                case 'LevelRule'
                    tempLevelRulesText = { tempLevelRulesText{1:end},
tempLevelRule };
                    inLevelRule = false;
                    tempLevelRule = {};
                case 'LevelRulesText'
                    tempSystem.LevelRulesText = tempLevelRulesText;
                    inLevelRulesText = false;
                    tempSystem.LevelRules = parseCell(tempLevelRulesText,ad);
                    tempLevelRulesText = {};
                case 'Cell'
                    if (inLevelAnd)
                        tempLevelAnd = { tempLevelAnd{1:end}, tempCell };
                    end
                otherwise
                    if (inShipLoad)
                        eval(sprintf('tempLoad.%s = tempCell;',myQueue{1}));
                    elseif (inShipSystem)
                        eval(sprintf('tempSystem.%s =
tempCell;',myQueue{1}));
                    end
                    inGroup = false;
```

```
            end
            if (numel(myQueue)>1)
                myQueue = { myQueue{2:end} };
            else
                myQueue = {};
            end
        case '};'
            break;
        otherwise
            if strcmp(val,'{')
                inGroup = true;
                myQueue = { PropName, myQueue{1:end} };
            else % Property of ShipSystem of ShipLoad
                if (inShipLoad)
                    eval(sprintf('tempLoad.%s = val;',PropName));
                elseif (inShipSystem)
                    eval(sprintf('tempSystem.%s = val;',PropName));
                end
            end
    end
end
fclose(fid);
```